GTP User Guide for Titan2 Series Products

(UG050007, V1.0)

(2021.6.20)

Document version revision record

Version number	Release date	Revision history
V0.1	2020/9/27	First draft
V0.2	2020/12/30	1.Added GTP_HPIO_VREF in Section (21) of Chapter 5; 2.Chapter 5 Section (14)GTP_IOBUFCO parameter list (Titan2 HPIO) update; 3.Chapter 5 Section (18)GTP_IOBUFE parameter list (Titan2) update; 4.Chapter 5 Section (19)GTP_IOBUFECO parameter list (Titan2HPIO) update. 5.Chapter 5 Section (16)GTP_IOBUF_RX_MIPI parameter list added parameter HPIO; 6.Chapter 5 Section (14)GTP_DLL_E2 parameter list added parameter HPIO; 7.Chapter 5 Section (15)GTP_CLKPD parameter list adds a new parameter HPIO, CPD_EDGE, which can be configured as rising edge alignment and falling edge alignment; 8.Chapter 5 Section (5)GTP_OSERDES_E2 Increase or decrease some ports and parameters; 9.Chapter 7 Section (6)GTP_IODELAY_E2 port list modify the port name; 10.Chapter 7 Section (19)GTP_RES_CAL_E1 Add and modify some ports and parameters; 11.Chapter 7 Section (22)GTP_HSSTHP_LANE Increase or decrease some ports and parameters, modify the default values of some parameters; 12.Chapter 7 Section (23) GTP_HSSTHP_HPLL parameter list Modify the default value of some parameters and name.
V0.3	2021/2/2	1.Modify the value range of parameters ALMOST_FULL_OFFSET / ALMOST_EMPTY_OFFSET in the GTP_FIFO36K_E1 parameter list in Section (18) and the. GTP_FIFO18K_E1 parameter list in Section (19) of Chapter 4; 2.The former Chapter 5, Section (21) GTP_HPIO_VREF is changed to Chapter 7, Section (24); 3.Added GTP_IOBUFCO_E1 to Section (21) of Chapter 5; 4.Added GTP_OUTBUFCO_E1 in Section (22) of Chapter 5; 5.Added GTP_OUTBUFTCO_E1 in Section (23) of Chapter 5; 6.Section (11) of Chapter Seven is removed GTP_IPAL_E1; 7.Chapter Seven, Section (18) GTP_RES_CAL_E1 parameter list added coordinate description, BANK_LOC parameter.
V0.4	2021/2/18	1.Chapter 7 Section (5) GTP_OSERDES_E2 parameter list the parameter TERM_OFF_SET adds a description that some value settings are not currently supported; 2.Chapter 7, Section (13) GTP_UDID add timing diagram and remarks; 3.Chapter 7, Section (20) GTP_PCIEGEN3 modify structure diagram; 4.Chapter 7, Section (21)GTP_HSSTHP_LANE add specific low-speed scene description, modify instantiation template; 5.Chapter 7 Section (22)GTP_HSSTHP_HPLL Modify the function description of theport list and merge the function description of the parameter list.
V0.5	2021/3/16	1.New chapterGTP classification; 2.Chapter 6 Section (3) GTP_CLKBUFGMUX instruction update; 3.Original Chapter 7 GTP_ISERDES_E2, GTP_OSERDES_E2, GTP_IODELAY_E2, GTP_ZEROHOLDDELAY nstructions are modified to Chapter 5, Section (24) ~ Section (27) for description, and some instructions are added at the same time; 4.Original Chapter 7 GTP_OSC_E4 is changed to Chapter 6 (15) for description; 5.Added chapter 7 configuration-related, GTP_EFUSECODE, GTP_IPAL_E2, GTP_SCANCHAIN_E1, GTP_UDID, GTP_JTAGIF, GTP_KEYRAM,

		GTP_CFGCLK instructions are modified to the current Chapter 7 Configuration Section (1) ~ Section (7); 6.Added Chapter 8 DDR-related GTP Instructions, formerly Chapter 6 GTP_IOCLKDIV_E3, GTP_CLKPD, GTP_DDC_E2 instructions for use, formerly Chapter 7 GTP_IDDR_E1, GTP_ODDR_E1 Instructions for use changed to Chapter 8 Section (1) ~ Section (5).
V1.0	2021/6/20	1.GTP classification chapter added description of GTP; 2.Chapter Five Section (1) ~ Section (6) GTP_INBUF instruction.GTP_INBUFG instruction, GTP_INBUFDS illustrate instruction, GTP_INBUFGDS instruction, GTP_INBUFE instruction, GTP_INBUFEDS update the list of valid values for parameters with instructions; 3.Modify the description of the port tri-state enable input signal T in Chapter 5; 4.Chapter 8 Section (2) GTP_CLKPD parameter list parameter HPIO "FALSE"; 5.Chapter 9 Section (7) GTP_HSSTHP_LANE, Section (8) GTP_HSSTHP_HPLL Section (17) Use instructions to update port, parameter and instantiation template; 6.Add Chapter 9, Section (9) GTP_HSSTHP_BUFDS instruction; 7.Chapter 4, Section (16) GTP_DRM36K_E1 parameter list. Section (17)GTP_DRM18K_E1 parameter list modification part description; 8.Chapter Five, Section (18) ~ Section (20) GTP_IOBUFE parameter list, GTP_IOBUFECO parameter list, GTP_IOBUFEDS parameter list remove parameters HPIO; 9.Chapter 5, section (22) GTP_OUTBUFCO_E1 of structure diagram, port list, section (23) GTP_OUTBUFTCO_E1 ofstructural block diagram, port list modify the wrong port name; 10.Chapter 5 Section (25) GTP_OSERDES_E2 parameter list renew; 11.Chapter 9 Section (3)GTP_START_E1 port list fixed wrong port names; 12.Chapter 9, Section (6)GTP_PCIEGEN3 port list delete wrong ports and add missing ports; 13.Chapter 9, Section (10)GTP_HPIO_VREF parameter list new parameter BANK_LOC.

Glossary

Abbreviations	Full Spelling
GTP	Generic Technology Primitive
FPGA	Field Programmable Gate Array
DRM	Dedicated RAM Module
HSST	High Speed Serial Transceiver
APM	Arithmetic Process Module
SRB	Signal Relay Block
ADC	Analog to Digital Converter
HRIO	High range IO
HPIO	High peformance IO

Table of contents

Table list

Figure list

GTP classification

GTP (Generic Technology Primitive, general technical primitive) is the name of a series of general-purpose modules developed for the characteristics of our company's devices. It is the basic unit in the chip and represents the hardware logic unit actually owned by the FPGA, such as LUT, DFF, DRM, etc.

This chapter divides the GTP primitives described in this article into nine categories according to their functions: flip-flops, latches, LUT-related, ROM and RAM-related, IO-related, clock-related, configuration-related, DDR-related, and other nine categories. See the table below for specific classification.

Table 1 Titan2 series GTP classification table

GTP name	GTP description
Triggers	
GTP_DFF	Trigger
GTP_DFF_C	Asynchronous Clear Flip-Flop
GTP_DFF_E	Flip-Flop with Enable
GTP_DFF_CE	Asynchronous Clear Flip-Flop with Enable
GTP_DFF_P	Asynchronous Set Flip-Flop
GTP_DFF_PE	Asynchronous Set Flip-Flop with Enable
GTP_DFF_R	Synchronous Clear Flip-Flops
GTP_DFF_RE	Synchronous Clear Flip-Flop with Enable
GTP_DFF_S	Synchronous Set Flip-Flops
GTP_DFF_SE	Synchronous Set Flip-Flop with Enable
Latches	
GTP_DLATCH	Latches
GTP_DLATCH_E	Latch with enable
GTP_DLATCH_C	Asynchronously clear the latch
GTP_DLATCH_CE	Asynchronous clear latch with enable
GTP_DLATCH_P	Asynchronous Set Latch
GTP_DLATCH_PE	Asynchronous Set Latch with Enable
LUT related	
GTP_ONE	GTP_ONE output high level
GTP_ZERO	GTP_ZERO output low level
GTP_BUF	Universal 1bit BUF
GTP_INV	Inverter
GTP_LUT1	1 input LUT
GTP_LUT2	2 input LUTs
GTP_LUT3	3 input LUTs
GTP_LUT4	4 input LUTs
GTP_LUT5	5 input LUTs
GTP_LUT6	6 input LUTs
GTP_LUT7	7 Input LUTs
GTP_LUT8	8 input LUTs
GTP_LUT6CARRY	6-input LUT with carry chain
GTP_LUT6D	6-input LUT with 2 output ports
GTP_MUX2LUT7	GTP_LUT7 dedicated two-input selector
GTP_MUX2LUT8	GTP_LUT8 dedicated two-input selector

GTP name	GTP description
ROM and RAM related	
GTP_ROM32X1	32bits address 1bit data ROM
GTP_ROM32X2	32bits address 2bits data ROM
GTP_ROM64X1	64bits address 1bit data ROM
GTP_ROM128X1	128bits address 1bit data ROM
GTP_ROM256X1	256bits address 1bit data ROM
GTP_RAM32X1SP	32bits address 1bit data single port RAM
GTP_RAM32X1DP	32bits address 1bit data simple dual port RAM
GTP_RAM32X2SP	32bits address 2bits data single port RAM
GTP_RAM32X2DP	32bits address 2bits data simple dual port RAM
GTP_RAM64X1SP	64bits address 1bit data single port RAM
GTP_RAM64X1DP	64bits address 1bit data simple dual port RAM
GTP_RAM128X1SP	128bits address 1bit data single port RAM
GTP_RAM128X1DP	128bits address 1bit data simple dual port RAM
GTP_RAM256X1SP	256bits address 1bit data single port RAM
GTP_DRM36K_E1	36K bits dedicated RAM module (DRM)
GTP_DRM18K_E1	18K bits dedicated RAM module (DRM)
GTP_FIFO36K_E1	36K bits FIFO
GTP_FIFO18K_E1	18K bits FIFO
IO-related	
GTP_INBUF	Single-ended input BUFFER
GTP_INBUFG	Single-ended input clock BUFFER
GTP_INBUFDS	Differential input BUFFER
GTP_INBUFGDS	Differential input clock BUFFER
GTP_INBUFE	Single-Ended Input BUFFER with Enable
GTP_INBUFEDS	Differential input BUFFER with enable terminal
GTP_OUTBUF	Single-ended output BUFFER
GTP_OUTBUFT	Three-state output BUFFER
GTP_OUTBUFDS	True differential output BUFFER
GTP_OUTBUFTDS	Three-state true differential output BUFFER
GTP_OUTBUFCO	Pseudo differential output BUFFER
GTP_OUTBUFTCO	Three-state pseudo-differential output BUFFER
GTP_IOBUF	Bidirectional input and output BUFFER
GTP_IOBUFCO	Bidirectional pseudo-differential input and output BUFFER
GTP_IOBUFDS	Bi-directional true differential input and output BUFFER
GTP_IOBUF_RX_MIPI	Support MIPI DPHY high-speed input, and single-ended input and output in low power mode (LP)
GTP_IOBUF_TX_MIPI	Support MIPI DPHY tri-state output, including input/output in low power mode (LP) and differential output in high speed (HS) mode
GTP_IOBUFE	Bidirectional input and output BUFFER with enable terminal
GTP_IOBUFECO	Bidirectional pseudo-differential input and output BUFFER with enable port
GTP_IOBUFEDS	Bidirectional True Differential Output BUFFER with Enable
GTP_IOBUFCO_E1	Bidirectional pseudo-differential input and output BUFFER
GTP_OUTBUFCO_E1	Pseudo differential input BUFFER
GTP_OUTBUFTCO_E1	Three-state pseudo-differential output BUFFER
GTP_ISERDES_E2	Input direction data processing module
GTP_OSERDES_E2	Data output conversion module

GTP name	GTP description
GTP_IODELAY_E2	Data delay unit, support dynamic or static delay control
GTP_ZEROHOLDDELAY	Data delay unit, supports static delay control
Clock related	
GTP_CLKBUFG	Global Clock BUFFER
GTP_CLKBUFGCE	Global clock BUFFER with enable terminal
GTP_CLKBUFGMUX	Dynamic switching GTP between two global clock input sources
GTP_CLKBUFR	Regional Clock BUFFER
GTP_IOCLKBUF	Equivalent to the gate function, the GTP can be used to turn off or turn on the IO clock
GTP_IOCLKDIV_E2	Clock Divide GTP
GTP_CLKBUFMCE	Clock buffers spanning clock regions (vertically) with enable
GTP_CLKBUFM	Clock Buffers Across Clock Regions (Vertical Direction)
GTP_CLKBUFXCE	Clock buffers spanning clock regions (horizontally) with enable
GTP_CLKBUFX	Clock buffers across clock regions (horizontally)
GTP_GPLL	GPLL is the core subsystem that FPGA provides clock resources
GTP_PPLL	PPLL is the core subsystem that FPGA provides clock resources
GTP_DLL_E2	Dynamically lock the frequency of the input reference clock, and output the equivalent delay step number
GTP_OSC_E4	Output fixed 50MHz clock to CCS as system clock
Configuration related	
GTP_EFUSECODE	The GTP is used to read EFUSECODE, and output 32bit data stored in efuse to the user in parallel
GTP_IPAL_E2	Used to implement CRC detection or SEU detection on the read-back data
GTP_SCANCHAIN_E1	Through the JTAG interface, read the value of the user data register
GTP_UDID	This GTP is used to read the UDID CODE value
GTP_JTAGIF	Implements user JTAG interface functionality
GTP_KEYRAM	Clear internal memory
GTP_CFGCLK	The GTP is the FLASH clock and clock enable sent to CCS by the SRB
DDR related	
GTP_IOCLKDIV_E3	Divide the clock from the PLL IOCLK clock tree to generate the system clock for DDRPHY
GTP_CLKPD	Detect the phase difference between MEMORY CONTROLLER soft core clock and DDR_PHY system clock, and output FLAG_PD signal to SRB
GTP_DDC_E2	Generate write operation clock for DDR memory
GTP_IDDR_E1	Data deserializer
GTP_ODDR_E1	Data parallel-to-serial processing module
Others	
GTP_APM_E2	logic operation unit
GTP_GRS	The GTP is used to control the global reset signal
GTP_START_E1	This GTP is used to describe the global signal release process for wakeup operations.
GTP_ADC_E2	Used to implement the ADC function
GTP_RES_CAL_E1	Impedance calibration for HPIO when describing user mode
GTP_PCIEGEN3	High-speed serial interface module, compatible with PCIE3.0 protocol
GTP_HSSTHP_LANE	1~4 full-duplex transceiver LANE supported by each HSSTHP (high-performance high-speed serial transceiver)
GTP_HSSTHP_HPLL	GTP_HSSTHP_HPLL is a dedicated Hpll module for HSSTHP (High Performance High Speed Serial Transceiver)
GTP_HSSTHP_BUFDS	Inside HSSTHP, each Quad has 2 HSSTHP_BUFDS
GTP_HPIO_VREF	Responsible for providing reference voltage to GPIO

1. Trigger GTP Instructions

(1) General introduction

GTP_DFF series GTP mainly implements trigger-related functions.

(2) GTP_DFF instructions

1. Supported devices

Table 2 GTP DFF Device Support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP DFF is a D flip-flop with data input D and data output Q, triggering on the rising edge of the clock, passing the input signal to the output signal. Power-on output is low.

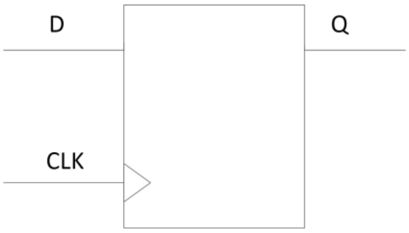


Figure 1 GTP DFF

3. Port description

Table 3 GTP DFF Port Descriptions

Port Name	Direction	Functional description
CLK	Input	Input clock
D	Input	Trigger input signal
Q	Output	Trigger output signal

4. Parameter description

Table 4 GTP_DFF parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b0	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiate templates

```
GTP_DFF #(
```

```
.GRS_EN("TRUE"),
.INIT(1'b0)
) GTP_DFF_inst (
.Q  (qout),
.CLK(clk),
.D  (d)
);
```

(3) GTP_DFF_C Instructions

1. Supported devices

Table 5 GTP_DFF_C device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DFF_C is a D flip-flop with data input D and data output Q, triggered on the rising edge of the clock, passing the input signal to the output signal. Asynchronous clear.

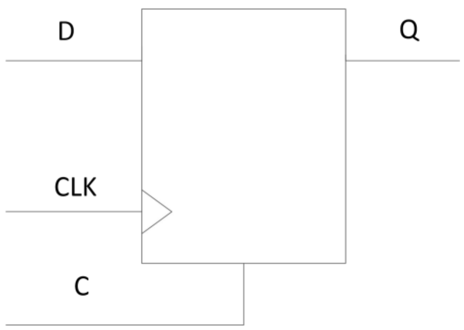


Figure 2 GTP_DFF_C

3. Port description

Table 6 GTP_DFF_C port description

Port Name	Direction	Functional description
CLK	Input	Input clock
D	Input	Trigger input signal
Q	Output	Trigger output signal
C	Input	Asynchronous clear signal

4. Parameter description

Table 7 GTP_DFF_C parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b0	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiate templates

```
TP_DFF_C #(
.GRS_EN("TRUE"),
.INIT(1'b0)
) GTP_DFF_C_inst (
.Q(qout),
.C(clear),
.CLK(clk),
.D(d)
);
```

(4) GTP_DFF_E instructions

1. Supported devices

Table 8 GTP_DFF_E device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DFF_E is a D flip-flop with data input D and data output Q, triggering on the rising edge of the clock and CE is high, passing the input signal to the output signal.

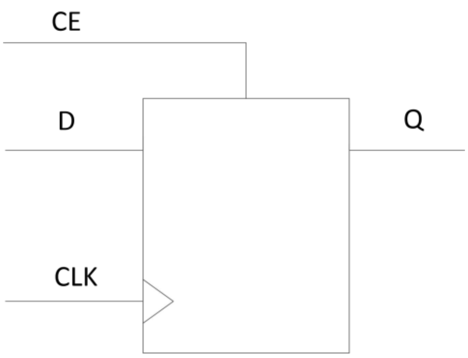


Figure 3 GTP_DFF_E

3. Port description

Table 9 GTP_DFF_E port description

Port Name	Direction	Functional description
CLK	Input	Input clock
D	Input	Trigger input signal
Q	Output	Trigger output signal
CE	Input	Enable signal, active high

4. Parameter description

Table 10 GTP_DFF_E parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b0	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiate templates

```
GTP_DFF_E #(
.GRS_EN("TRUE"),
.INIT(1'b0)
) GTP_DFF_E_inst (
.Q(q),
.CE(ce),
.CLK(clk),
.D(d)
);
```

(5) GTP_DFF_CE Instructions

1. Supported devices

Table 11 GTP_DFF_CE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DFF_CE is a D flip-flop with data input D and data output Q. When the rising edge of the clock is triggered and CE is high, the input signal is passed to the output signal. Asynchronously cleared.

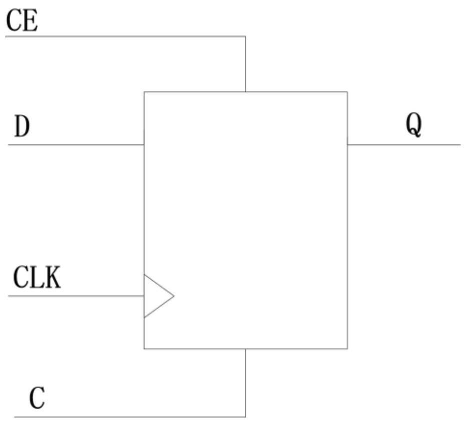


Figure 4 GTP_DFF_CE

3. Port description

Table 12 GTP_DFF_CE port description

Port Name	Direction	Functional description
CLK	Input	Input Clock
D	Input	Trigger input signal
Q	Output	Trigger output signal
C	Input	Asynchronous clear signal
CE	Input	Enable signal, highly effective

4. Parameter description

Table 13 GTP_DFF_CE parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b0	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DFF_CE #(
.GRS_EN("TRUE"),
.INIT(1'b0)
) GTP_DFF_CE_inst (
.Q             (q ),
.C             (c ),
.CE            (ce ),
.CLK           (clk),
.D             (d )
);
```

(6) GTP_DFF_P instruction

1. Supported devices

Table 14 GTP_DFF_P device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DFF_P is a D flip-flop with data input D and data output Q, which is triggered on the rising edge of the clock and transmits the input signal to the output signal. Asynchronous setting.

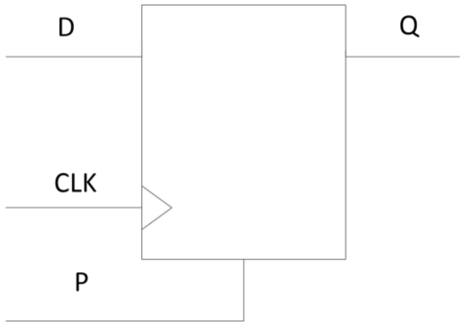


Figure 5 GTP_DFF_P

3. Port description

Table 15 GTP_DFF_P port description

Port Name	Direction	Functional description
CLK	Input	Input Clock
D	Input	Trigger input signal
Q	Output	Trigger output signal
P	Input	Asynchronous setting signal

4. Parameter description

Table 16 GTP_DFF_P parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b1	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DFF_P #(
.GRS_EN("TRUE"),
.INIT(1'b1)
) GTP_DFF_P_inst (
.Q(q),
.CLK(clk),
.D(d),
.P(p)
);
```

(7) GTP_DFF_PE instructions

1. Supported devices

Table 17 GTP_DFF_PE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DFF_PE is a D flip-flop with data input D and data output Q. When CE is high, it is triggered on the rising edge of the clock to pass the input signal to the output signal. Asynchronous setting.

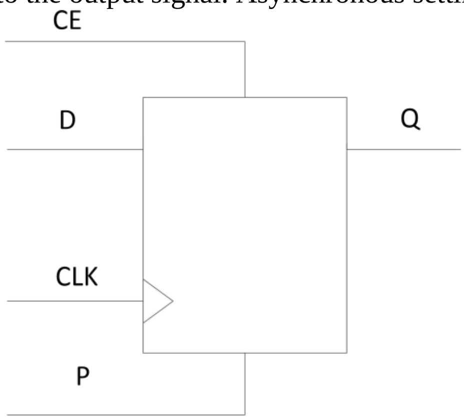


Figure 6 GTP_DFF_PE

3. Port description

Table 18 GTP_DFF_PE port description

Port Name	Direction	Functional description
CLK	Input	Input Clock
D	Input	Trigger input signal
Q	Output	Trigger output signal
P	Input	Asynchronous setting signal
CE	Input	Enable signal, highly effective

4. Parameter description

Table 19 GTP_DFF_PE parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b1	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DFF_PE #(
.GRS_EN("TRUE"),
.INIT(1'b1)
) GTP_DFF_PE_inst (
.Q(q),
.CE(ce),
.CLK(clk),
.D(d),
.P(p)
);
```

(8) GTP_DFF_R instruction

1. Supported devices

Table 20 GTP_DFF_R device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DFF_R is a D flip-flop with data input D and data output Q, triggered on the rising edge of the clock, passing the input signal to the output signal. Synchronously cleared.

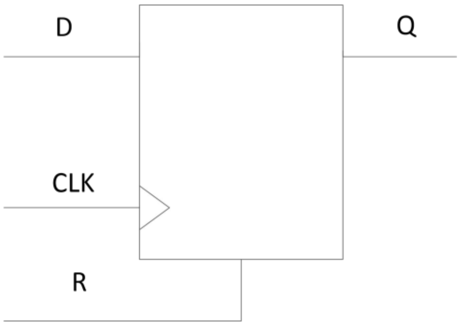


Figure 7 GTP_DFF_R

3. Port description

Table 21 GTP_DFF_R port description

Port Name	Direction	Functional description
CLK	Input	Input Clock
D	Input	Trigger input signal
Q	Output	Trigger output signal
R	Input	Synchronous clear signal

4. Parameter description

Table 22 GTP_DFF_R parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b0	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DFF_R #(
.GRS_EN("TRUE"),
.INIT(1'b0)
) GTP_DFF_R_inst (
.Q(q),
.CLK(clk),
```

```
.D(d),
.R(r)
);
```

(9) GTP_DFF_RE instruction

1. Supported devices

Table 23 GTP_DFF_RE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DFF_RE is a D flip-flop with data input D and data output Q. When CE is high, it triggers on the rising edge of the clock, passing the input signal to the output signal. Synchronous clear.

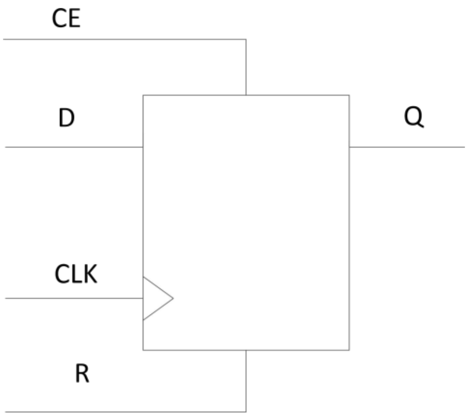


Figure 8 GTP_DFF_RE

3. Port description

Table 24 GTP_DFF_RE port description

Port Name	Direction	Functional description
CLK	Input	Input clock
D	Input	Trigger input signal
Q	Output	Trigger output signal
R	Input	Synchronous clear signal
CE	Input	Enable signal, highly effective

4. Parameter description

Table 25 GTP_DFF_RE parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b0	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DFF_RE #(
.GRS_EN("TRUE"),
.INIT(1'b0)
) GTP_DFF_RE_inst (
.Q(q),
.CE(ce),
.CLK(clk),
.D(d),
.R(r)
);
```

(10) GTP_DFF_S instruction

1. Supported devices

Table 26 GTP_DFF_S device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DFF_S is a D flip-flop with data input D and data output Q, triggered on the rising edge of the clock, passing the input signal to the output signal. Synchronously set.

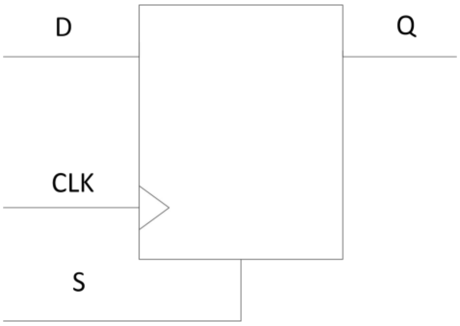


Figure 9 GTP_DFF_S

3. Port description

Table 27 GTP_DFF_S port description

Port Name	Direction	Functional description
CLK	Input	Input clock
D	Input	Trigger input signal
Q	Output	Trigger output signal
S	Input	Synchronous set signal

4. Parameter description

Table 28 GTP_DFF_S parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b1	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DFF_S #(
.GRS_EN("TRUE"),
.INIT(1'b1)
) GTP_DFF_S_inst (
.Q(q),
.CLK(clk),
.D(d),
.S(s)
);
```

(11) GTP_DFF_SE instruction

1. Supported devices

Table 29 GTP_DFF_SE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DFF_SE is a D flip-flop with data input D and data output Q, triggering on the rising edge of the clock and CE is high, passing the input signal to the output signal. Synchronously set.

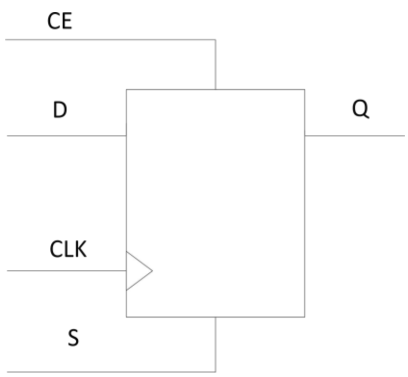


Figure 10 GTP_DFF_SE

3. Port description

Table 30 GTP_DFF_SE port description

Port Name	Direction	Functional description
CLK	Input	Input clock
D	Input	Trigger input signal
Q	Output	Trigger output signal
S	Input	Synchronous set signal
CE	Input	Enable signal, highly effective

4. Parameter description

Table 31 GTP_DFF_SE parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b1	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DFF_SE #(
.GRS_EN("TRUE"),
.INIT(1'b1)
) GTP_DFF_SE_inst (
.Q(q),
.CE(ce),
.CLK(clk),
.D(d),
.S(s)
);
```

2. Latches GTP Instructions

(1) General introduction

GTP_DLATCH series GTP mainly implements latch-related functions.

(2) GTP_DLATCH instruction

1. Supported devices

Table 32 GTP_DLATCH Device Support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DLATCH is a latch with data input D and data output Q. When G is high, it passes the input signal to the output signal.

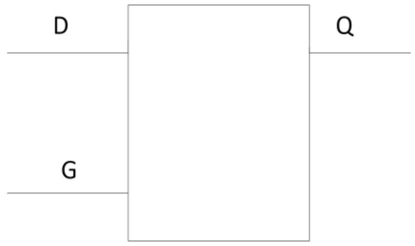


Figure 11 Schematic diagram of GTP_DLATCH

3. Port description

Table 33 GTP_DLATCH port list

Port Name	Direction	Functional description
G	Input	Level input
D	Input	Latch input signal
Q	Output	Latch output signal

4. Parameter description

Table 34 GTP_DLATCH parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b0	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DLATCH#(
.INIT(1'b0),
.GRS_EN("TURE")//"TURE"; "FALSE"
)
GTP_DLATCH_inst (
.Q (q),
.D (d),
.G (g)
);
```

(3) GTP_DLATCH_E instruction

1. Supported devices

Table 35 GTP_DLATCH_E device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DLATCH_E is a latch with data input D and data output Q, when G is high and GE is high, the input signal is passed to the output signal.

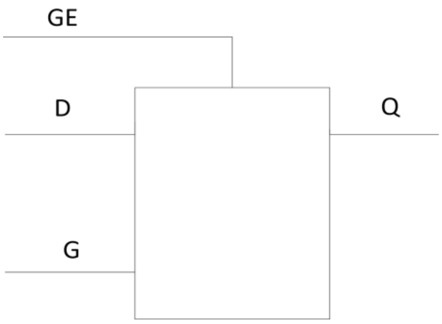


Figure 12 GTP_DLATCH_E schematic diagram

3. Port description

Table 36 GTP_DLATCH_E port list

Port Name	Direction	Functional description
G	Input	Input level
D	Input	Latch Input Signal
Q	Output	Latch output signal
GE	Input	Enable signal, highly effective

4. Parameter description

Table 37 GTP_DLATCH_E parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b0	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DLATCH_E#(
.INIT (1'b0),
.GRS_EN("TURE")//"TURE"; "FALSE"
)
GTP_DLATCH_E_inst (
.Q (q),
.D (d),
.G (g),
.GE (en)
);
```

(4) GTP_DLATCH_C instruction

1. Supported devices

Table 38 GTP_DLATCH_C device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DLATCH_C is a latch with data input D and data output Q. When G is high, the input signal is passed to the output signal. Asynchronously cleared.

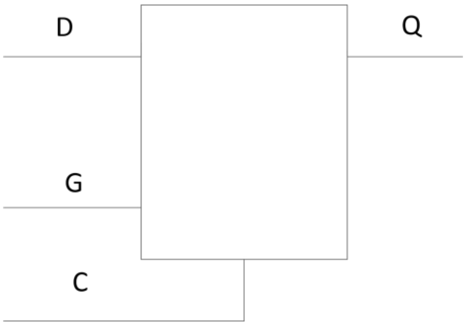


Figure 13 GTP_DLATCH_C schematic diagram

3. Port description

Table 39 GTP_DLATCH_C port list

Port Name	Direction	Functional description
G	Input	Input level
D	Input	Latch input signal
Q	Output	Latch output signal
C	Input	Asynchronous clear signal

4. Parameter description

Table 40 GTP_DLATCH_E parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b0	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DLATCH_C#(
.INIT(1'b0),
.GRS_EN("TURE")//"TURE"; "FALSE"
)
GTP_DLATCH_C_inst (
.Q (q),
.D (d),
.G (g),
.C (c)
);
```

(5) GTP_DLATCH_CE instruction

1. Supported devices

Table 41 GTP_DLATCH_CE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DLATCH_CE is a latch with data input D and data output Q. When G is high and GE is high, the input signal is passed to the output signal. Asynchronously cleared.

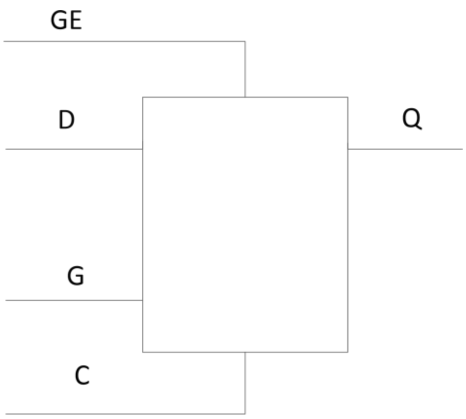


Figure 14 GTP_DLATCH_CE schematic diagram

3. Port description

Table 42 GTP_DLATCH_CE port list

Port Name	Direction	Functional description
G	Input	Input level
D	Input	Latch input signal
Q	Output	Latch output signal
C	Input	Asynchronous clear signal
GE	Input	Enable signal, highly effective

4. Parameter description

Table 43 GTP_DLATCH_CE parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b0	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DLATCH_CE#(
.INIT(1'b0),
.GRS_EN("TURE")//"TURE"; "FALSE"
)
GTP_DLATCH_CE_inst (
.Q (q),
.D (d),
.G (g),
.C (c),
.GE (ge)
);
```

(6) GTP_DLATCH_P instruction

1. Supported devices

Table 44 GTP_DLATCH device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DLATCH_P is a latch with data input D and data output Q. When G is high, the input signal is passed to the output signal. Asynchronous set.



Figure 15 GTP_DLATCH_P schematic diagram

3. Port description

Table 45 GTP_DLATCH_P port list

Port Name	Direction	Functional description
G	Input	Input level
D	Input	Latch input signal
Q	Output	Latch output signal
P	Input	Asynchronous setting signal

4. Parameter description

Table 46 GTP_DLATCH_CE parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b1	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DLATCH_P#(
.INIT(1'b1),
.GRS_EN("TURE")//"TURE"; "FALSE"
)
GTP_DLATCH_P_inst (
```

```
.Q (q),
.D (d),
.G (g),
.P (p)
);
```

(7) GTP_DLATCH_PE instruction

1. Supported devices

Table 47 GTP_DLATCH device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DLATCH_PE is a latch with data input D and data output Q, when G is high and GE is high, the input signal is passed to the output signal. Set asynchronously.

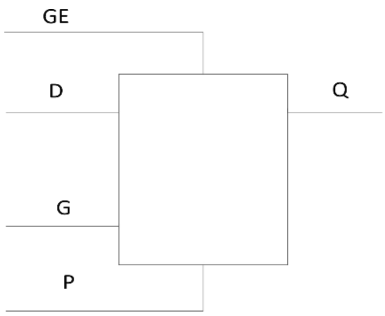


Figure 16 GTP_DLATCH_PE schematic diagram

3. Port description

Table 48 GTP_DLATCH_PE port list

Port Name	Direction	Functional description
G	Input	Input level
D	Input	Latch input signal
Q	Output	Latch output signal
P	Input	Asynchronous setting signal
GE	Input	Enable signal, highly effective

4. Parameter description

Table 49 GTP_DLATCH_CE parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	"TRUE", "FALSE"	"TRUE"	Global reset signal enable
INIT	<binary>	1'b0, 1'b1	1'b1	Initial value, triggered asynchronously by the global reset signal (this parameter simulation does not support)

5. Instantiation template

```
GTP_DLATCH_PE#(
.INIT(1'b1),
.GRS_EN("TURE")//"TURE"; "FALSE"
)
GTP_DLATCH_PE_inst (
(q),
.D (d),
.G (g),
.P (p),
.GE (en)
);
```

3. LUT related GTP instructions

(1) GTP_ONE instruction

1. Supported devices

Table 50 GTP_ONE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_ONE outputs high level. As shown in the figure below:

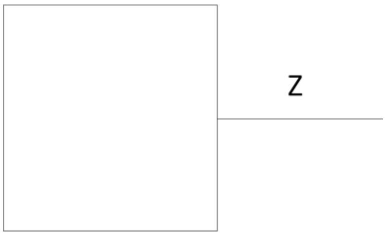


Figure 17 GTP_ONE structure diagram

3. Port description

Table 51 GTP_ONE port list

Port Name	Direction	Functional description
Z	Output	High level output

4. Instantiation template

```
GTP_ONE GTP_ONE_inst (
.Z (Z)
);
```

(2) GTP_ZERO instruction

1. Supported devices

Table 52 GTP_ZERO device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_ZERO outputs low level. The structure diagram is as follows:

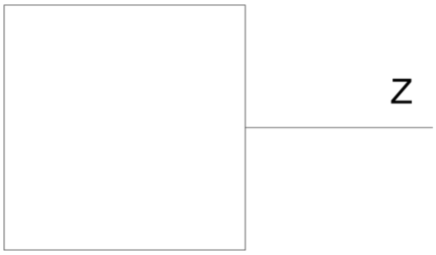


Figure 18 GTP_ZERO structure diagram

3. Port description

Table 53 GTP_ZERO port list

Port Name	Direction	Functional description
Z	Output	Low level output

4. Instantiation template

```
GTP_ZERO GTP_ZERO_inst (
.Z   (z)
);
```

(3) GTP_BUF instruction

1. Supported devices

Table 54 GTP_BUF device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_BUF is a general 1bit BUFFER, the structure diagram is as follows:

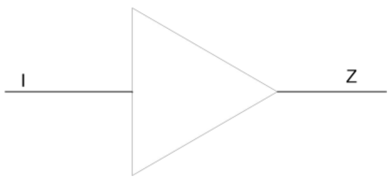


Figure 19 GTP_BUF structure diagram

3. Port description

Table 55 GTP_BUF port list

Port Name	Direction	Functional description
Z	Output	Output signal
I	Input	Input signal

4. Instantiation template

```
GTP_BUF GTP_BUF_inst (
.I   (I),
.Z   (Z)
);
```

5. Detailed function description

This GTP is not necessary, the software will optimize GTP_BUF after MAP.

(4) GTP_INV instruction

1. Supported devices

Table 56 GTP_INV device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_INV is an inverter. The structure diagram is as follows:

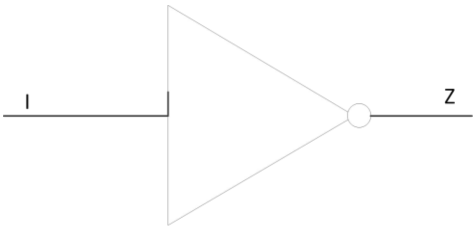


Figure 20 GTP_INV structure diagram

3. Port description

Table 57 GTP_INV port list

Port Name	Direction	Functional description
Z	Output	Inverse output signal
I	Input	Input signal

4. Instantiation template

```
GTP_INV  GTP_INV_inst (
.I   (i),
.Z   (z)
);
```

(5) GTP_LUT1 instruction

1. Supported devices

Table 58 GTP_LUT1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_LUT1 is a lookup table with 1bit input and 1bit output. Its structural block diagram is as follows:

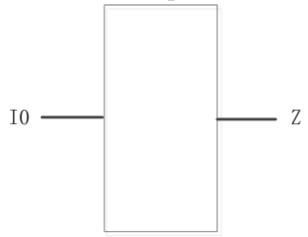


Figure 21 GTP_LUT1 structure diagram

3. Port description

Table 59 GTP_LUT1 port list

Port Name	Direction	Functional description
Z	Output	Output signal
I0	Input	Input signal

Truth table:

Table 60 GTP_LUT1 truth table

Input	Output
I0	Z
0	INIT[0]
1	INIT[1]

4. Parameter description

Table 61 GTP_LUT1 parameter list

Parameter Name	Describe	Defaults	Effective value
INIT	Output configuration parameters	2'h0	0~3

5. Instantiation template

```
GTP_LUT1 #(
.INIT           (2'h0)
)GTP_LUT1_inst (
.Z              (z),
.I0             (i0)
);
```

(6) GTP_LUT2 instruction

1. Supported devices

Table 62 GTP_LUT2 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_LUT2 is a two-input lookup table. Different INIT parameter values correspond to different functions of LUT2. When using it, you must specify the corresponding INIT parameter. Its structure diagram is as follows:

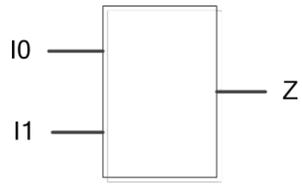


Figure 22 GTP_LUT2 structure diagram

3. Port description

Table 63 GTP_LUT2 port list

Port Name	Direction	Functional description
Z	Output	Output signal
I0	Input	Input signal
I1	Input	Input signal

Truth table:

Table 64 GTP_LUT2 truth table

Input		Output
I1	I0	Z
0	0	INIT[0]
0	1	INIT[1]
1	0	INIT[2]
1	1	INIT[3]

4. Parameter description

Table 65 GTP_LUT2 parameter list

Parameter Name	Describe	Defaults	Effective value
INIT	Output configuration parameters	4'h0	0~15

5. Instantiation template

```
GTP_LUT2 #(
.INIT         (4'h0)
)GTP_LUT2_inst (
.Z            (z),
.I0           (i0),
.I1           (i1)
);
```

(7) GTP_LUT3 instruction

1. Supported devices

Table 66 GTP_LUT3 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_LUT3 is a three-input lookup table. Different INIT parameter values correspond to different functions of LUT3. When using it, you must specify the corresponding INIT parameter. Its structure diagram is as follows：

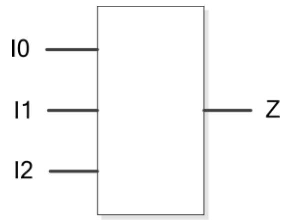


Figure 23 GTP_LUT3 structure diagram

3. Port description

Table 67 GTP_LUT3 port list

Port Name	Direction	Functional description
Z	Output	Output signal
I0	Input	Input signal
I1	Input	Input signal
I2	Input	Input signal

Truth table:

Table 68 GTP_LUT3 truth table

Input			Output
I2	I1	I0	Z
0	0	0	INIT[0]
0	0	1	INIT[1]
0	1	0	INIT[2]

Input			Output
0	1	1	INIT[3]
1	0	0	INIT[4]
1	0	1	INIT[5]
1	1	0	INIT[6]
1	1	1	INIT[7]

4. Parameter description

Table 69 GTP_LUT3 parameter list

Parameter Name	Describe	Defaults	Effective value
INIT	Output configuration parameters	8'h00	0~255

5. Instantiation template

```
GTP_LUT3 #(
.INIT          (8'h00)
)GTP_LUT3_inst (
.Z             (z),
.I0            (i0),
.I1            (i1),
.I2            (i2)
);
```

(8) GTP_LUT4 instruction

1. Supported devices

Table 70 GTP_LUT4 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_LUT4 is a 4-input lookup table. Different INIT parameter values correspond to different functions of LUT4. When using it, you must specify the corresponding INIT parameter. Its structure diagram is as follows：

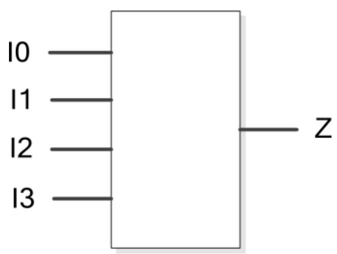


Figure 24 GTP_LUT4 structure diagram

3. Port description

Table 71 GTP_LUT4 port list

Port Name	Direction	Functional description
Z	Output	Output signal
I0	Input	Input signal
I1	Input	Input signal
I2	Input	Input signal
I3	Input	Input signal

Truth table:

Table 72 GTP_LUT4 truth table

Input				Output
I3	I2	I1	I0	Z
0	0	0	0	INIT[0]
0	0	0	1	INIT[1]
0	0	1	0	INIT[2]
0	0	1	1	INIT[3]
0	1	0	0	INIT[4]
0	1	0	1	INIT[5]
0	1	1	0	INIT[6]
0	1	1	1	INIT[7]
1	0	0	0	INIT[8]
1	0	0	1	INIT[9]
1	0	1	0	INIT[10]
1	0	1	1	INIT[11]
1	1	0	0	INIT[12]
1	1	0	1	INIT[13]
1	1	1	0	INIT[14]
1	1	1	1	INIT[15]

4. Parameter description

Table 73 GTP_LUT4 parameter list

Parameter Name	Describe	Defaults	Effective value
INIT	Output configuration parameters	16'h0000	0~2^16-1

5. Instantiation template

```
GTP_LUT4 #(
.INIT        (16'h0000)
)GTP_LUT4_inst (
.Z           (z),
.I0          (i0),
.I1          (i1),
.I2          (i2),
.I3          (i3)
);
```

(9) GTP_LUT5 instruction

1. Supported devices

Table 74 GTP_LUT5 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_LUT5 is a 5-input lookup table. Different INIT parameter values correspond to different functions of LUT5. When using it, you must specify the corresponding INIT parameter. Its structure diagram is as follows：

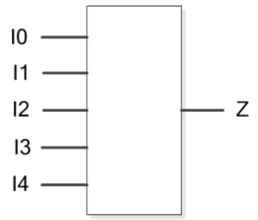


Figure 25 GTP_LUT5 structure diagram

3. Port description

Table 75 GTP_LUT5 port list

Port Name	Direction	Functional description
Z	Output	Output signal
I0	Input	Input signal
I1	Input	Input signal
I2	Input	Input signal
I3	Input	Input signal
I4	Input	Input signal

Truth table:

Table 76 GTP_LUT5 truth table

Input					Output
I4	I3	I2	I1	I0	Z
0	0	0	0	0	INIT[0]
0	0	0	0	1	INIT[1]
0	0	0	1	0	INIT[2]
0	0	0	1	1	INIT[3]
0	0	1	0	0	INIT[4]
0	0	1	0	1	INIT[5]
0	0	1	1	0	INIT[6]
0	0	1	1	1	INIT[7]
0	1	0	0	0	INIT[8]
0	1	0	0	1	INIT[9]
0	1	0	1	0	INIT[10]

Input					Output
0	1	0	1	1	INIT[11]
0	1	1	0	0	INIT[12]
0	1	1	0	1	INIT[13]
0	1	1	1	0	INIT[14]
0	1	1	1	1	INIT[15]
1	0	0	0	0	INIT[16]
1	0	0	0	1	INIT[17]
1	0	0	1	0	INIT[18]
1	0	0	1	1	INIT[19]
1	0	1	0	0	INIT[20]
1	0	1	0	1	INIT[21]
1	0	1	1	0	INIT[22]
1	0	1	1	1	INIT[23]
1	1	0	0	0	INIT[24]
1	1	0	0	1	INIT[25]
1	1	0	1	0	INIT[26]
1	1	0	1	1	INIT[27]
1	1	1	0	0	INIT[28]
1	1	1	0	1	INIT[29]
1	1	1	1	0	INIT[30]
1	1	1	1	1	INIT[31]

4. Parameter description

Table 77 GTP_LUT5 parameter

Parameter Name	Describe	Defaults	Settings
INIT	Output configuration parameters	32'h0000_0000	0 ~ 2^32-1

5. Instantiation template

```
GTP_LUT5 #(
.INIT          (32'h0000_0000)
)GTP_LUT5_inst (

.Z             (z),
.I0            (i0),
.I1            (i1),
.I2            (i2),
.I3            (i3),
.I4            (i4)
);
```

(10) GTP_LUT6 instruction

1. Supported devices

Table 78 GTP_LUT6 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_LUT6 is a 6-input lookup table. Different INIT parameter values correspond to different functions of LUT6. When using it, you must specify the corresponding INIT parameter. Its structure diagram is as follows:

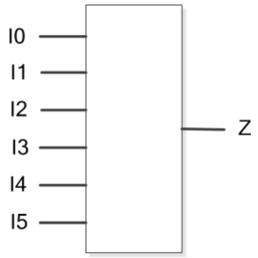


Figure 26 GTP_LUT6 structure diagram

3. Port description

Table 79 GTP_LUT6 port

Port Name	Direction	Functional description
Z	Output	Output signal
I0	Input	Input signal
I1	Input	Input signal
I2	Input	Input signal
I3	Input	Input signal
I4	Input	Input signal
I5	Input	Input signal

Truth table:

Table 80 GTP_LUT6 truth table

Input						Output
I5	I4	I3	I2	I1	I0	Z
0	0	0	0	0	0	INIT[0]
0	0	0	0	0	1	INIT[1]
0	0	0	0	1	0	INIT[2]
0	0	0	0	1	1	INIT[3]
0	0	0	1	0	0	INIT[4]
0	0	0	1	0	1	INIT[5]
0	0	0	1	1	0	INIT[6]
0	0	0	1	1	1	INIT[7]
0	0	1	0	0	0	INIT[8]

Input						Output
0	0	1	0	0	1	INIT[9]
0	0	1	0	1	0	INIT[10]
0	0	1	0	1	1	INIT[11]
0	0	1	1	0	0	INIT[12]
0	0	1	1	0	1	INIT[13]
0	0	1	1	1	0	INIT[14]
0	0	1	1	1	1	INIT[15]
0	1	0	0	0	0	INIT[16]
0	1	0	0	0	1	INIT[17]
0	1	0	0	1	0	INIT[18]
0	1	0	0	1	1	INIT[19]
0	1	0	1	0	0	INIT[20]
0	1	0	1	0	1	INIT[21]
0	1	0	1	1	0	INIT[22]
0	1	0	1	1	1	INIT[23]
0	1	1	0	0	0	INIT[24]
0	1	1	0	0	1	INIT[25]
0	1	1	0	1	0	INIT[26]
0	1	1	0	1	1	INIT[27]
0	1	1	1	0	0	INIT[28]
0	1	1	1	0	1	INIT[29]
0	1	1	1	1	0	INIT[30]
0	1	1	1	1	1	INIT[31]
1	0	0	0	0	0	INIT[32]
1	0	0	0	0	1	INIT[33]
1	0	0	0	1	0	INIT[34]
1	0	0	0	1	1	INIT[35]
1	0	0	1	0	0	INIT[36]
1	0	0	1	0	1	INIT[37]
1	0	0	1	1	0	INIT[38]
1	0	0	1	1	1	INIT[39]
1	0	1	0	0	0	INIT[40]
1	0	1	0	0	1	INIT[41]
1	0	1	0	1	0	INIT[42]
1	0	1	0	1	1	INIT[43]
1	0	1	1	0	0	INIT[44]
1	0	1	1	0	1	INIT[45]
1	0	1	1	1	0	INIT[46]
1	0	1	1	1	1	INIT[47]
1	1	0	0	0	0	INIT[48]
1	1	0	0	0	1	INIT[49]
1	1	0	0	1	0	INIT[50]
1	1	0	0	1	1	INIT[51]
1	1	0	1	0	0	INIT[52]
1	1	0	1	0	1	INIT[53]
1	1	0	1	1	0	INIT[54]

Input						Output
1	1	0	1	1	1	INIT[55]
1	1	1	0	0	0	INIT[56]
1	1	1	0	0	1	INIT[57]
1	1	1	0	1	0	INIT[58]
1	1	1	0	1	1	INIT[59]
1	1	1	1	0	0	INIT[60]
1	1	1	1	0	1	INIT[61]
1	1	1	1	1	0	INIT[62]
1	1	1	1	1	1	INIT[63]

4. Parameter description

Table 81 GTP_LUT6 parameter

Parameter Name	Describe	Defaults	Settings
INIT	Output configuration parameters	64'h0000_0000_0000_0000	0 ~ 2^64-1

5. Instantiation template

```
GTP_LUT6 #(
.INIT         (64'h0000_0000_0000_0000)
)GTP_LUT6_inst (

.Z            (z),
.I0           (i0),
.I1           (i1),
.I2           (i2),
.I3           (i3),
.I4           (i4),
.I5           (i5)
);
```

(11) GTP_LUT7 instruction

1. Supported devices

Table 82 GTP_LUT7 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_LUT7 is a 7-input lookup table. Different INIT parameter values correspond to different functions of LUT7. When using it, you must specify the corresponding INIT parameter. Its structure diagram is as follows:

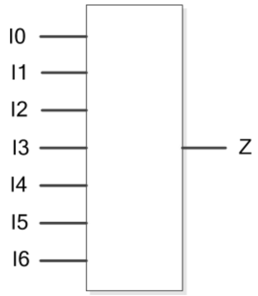


Figure 27 GTP_LUT7 structure diagram

3. Port description

Table 83 GTP_LUT7 port

Port Name	Direction	Functional description
Z	Output	Output signal
I0	Input	Input signal
I1	Input	Input signal
I2	Input	Input signal
I3	Input	Input signal
I4	Input	Input signal
I5	Input	Input signal
I6	Input	Input signal

Truth table:

Table 84 GTP_LUT7 truth table

Input							Output
I6	I5	I4	I3	I2	I1	I0	Z
0	0	0	0	0	0	0	INIT[0]
0	0	0	0	0	0	1	INIT[1]
0	0	0	0	0	1	0	INIT[2]
0	0	0	0	0	1	1	INIT[3]
0	0	0	0	1	0	0	INIT[4]
0	0	0	0	1	0	1	INIT[5]
0	0	0	0	1	1	0	INIT[6]
0	0	0	0	1	1	1	INIT[7]
0	0	0	1	0	0	0	INIT[8]
0	0	0	1	0	0	1	INIT[9]
0	0	0	1	0	1	0	INIT[10]
0	0	0	1	0	1	1	INIT[11]
0	0	0	1	1	0	0	INIT[12]
0	0	0	1	1	0	1	INIT[13]
0	0	0	1	1	1	0	INIT[14]
0	0	0	1	1	1	1	INIT[15]
0	0	1	0	0	0	0	INIT[16]
0	0	1	0	0	0	1	INIT[17]
0	0	1	0	0	1	0	INIT[18]
0	0	1	0	0	1	1	INIT[19]
0	0	1	0	1	0	0	INIT[20]

Intput							Output
0	0	1	0	1	0	1	INIT[21]
0	0	1	0	1	1	0	INIT[22]
0	0	1	0	1	1	1	INIT[23]
0	0	1	1	0	0	0	INIT[24]
0	0	1	1	0	0	1	INIT[25]
0	0	1	1	0	1	0	INIT[26]
0	0	1	1	0	1	1	INIT[27]
0	0	1	1	1	0	0	INIT[28]
0	0	1	1	1	0	1	INIT[29]
0	0	1	1	1	1	0	INIT[30]
0	0	1	1	1	1	1	INIT[31]
0	1	0	0	0	0	0	INIT[32]
0	1	0	0	0	0	1	INIT[33]
0	1	0	0	0	1	0	INIT[34]
0	1	0	0	0	1	1	INIT[35]
0	1	0	0	1	0	0	INIT[36]
0	1	0	0	1	0	1	INIT[37]
0	1	0	0	1	1	0	INIT[38]
0	1	0	0	1	1	1	INIT[39]
0	1	0	1	0	0	0	INIT[40]
0	1	0	1	0	0	1	INIT[41]
0	1	0	1	0	1	0	INIT[42]
0	1	0	1	0	1	1	INIT[43]
0	1	0	1	1	0	0	INIT[44]
0	1	0	1	1	0	1	INIT[45]
0	1	0	1	1	1	0	INIT[46]
0	1	0	1	1	1	1	INIT[47]
0	1	1	0	0	0	0	INIT[48]
0	1	1	0	0	0	1	INIT[49]
0	1	1	0	0	1	0	INIT[50]
0	1	1	0	0	1	1	INIT[51]
0	1	1	0	1	0	0	INIT[52]
0	1	1	0	1	0	1	INIT[53]
0	1	1	0	1	1	0	INIT[54]
0	1	1	0	1	1	1	INIT[55]
0	1	1	1	0	0	0	INIT[56]
0	1	1	1	0	0	1	INIT[57]
0	1	1	1	0	1	0	INIT[58]
0	1	1	1	0	1	1	INIT[59]
0	1	1	1	1	0	0	INIT[60]
0	1	1	1	1	0	1	INIT[61]
0	1	1	1	1	1	0	INIT[62]
0	1	1	1	1	1	1	INIT[63]
1	0	0	0	0	0	0	INIT[64]
1	0	0	0	0	0	1	INIT[65]
1	0	0	0	0	1	0	INIT[66]

Intput							Output
1	0	0	0	0	1	1	INIT[67]
1	0	0	0	1	0	0	INIT[68]
1	0	0	0	1	0	1	INIT[69]
1	0	0	0	1	1	0	INIT[70]
1	0	0	0	1	1	1	INIT[71]
1	0	0	1	0	0	0	INIT[72]
1	0	0	1	0	0	1	INIT[73]
1	0	0	1	0	1	0	INIT[74]
1	0	0	1	0	1	1	INIT[75]
1	0	0	1	1	0	0	INIT[76]
1	0	0	1	1	0	1	INIT[77]
1	0	0	1	1	1	0	INIT[78]
1	0	0	1	1	1	1	INIT[79]
1	0	1	0	0	0	0	INIT[80]
1	0	1	0	0	0	1	INIT[81]
1	0	1	0	0	1	0	INIT[82]
1	0	1	0	0	1	1	INIT[83]
1	0	1	0	1	0	0	INIT[84]
1	0	1	0	1	0	1	INIT[85]
1	0	1	0	1	1	0	INIT[86]
1	0	1	0	1	1	1	INIT[87]
1	0	1	1	0	0	0	INIT[88]
1	0	1	1	0	0	1	INIT[89]
1	0	1	1	0	1	0	INIT[90]
1	0	1	1	0	1	1	INIT[91]
1	0	1	1	1	0	0	INIT[92]
1	0	1	1	1	0	1	INIT[93]
1	0	1	1	1	1	0	INIT[94]
1	0	1	1	1	1	1	INIT[95]
1	1	0	0	0	0	0	INIT[96]
1	1	0	0	0	0	1	INIT[97]
1	1	0	0	0	1	0	INIT[98]
1	1	0	0	0	1	1	INIT[99]
1	1	0	0	1	0	0	INIT[100]
1	1	0	0	1	0	1	INIT[101]
1	1	0	0	1	1	0	INIT[102]
1	1	0	0	1	1	1	INIT[103]
1	1	0	1	0	0	0	INIT[104]
1	1	0	1	0	0	1	INIT[105]
1	1	0	1	0	1	0	INIT[106]
1	1	0	1	0	1	1	INIT[107]
1	1	0	1	1	0	0	INIT[108]
1	1	0	1	1	0	1	INIT[109]
1	1	0	1	1	1	0	INIT[110]
1	1	0	1	1	1	1	INIT[111]
1	1	1	0	0	0	0	INIT[112]

Input							Output
1	1	1	0	0	0	1	INIT[113]
1	1	1	0	0	1	0	INIT[114]
1	1	1	0	0	1	1	INIT[115]
1	1	1	0	1	0	0	INIT[116]
1	1	1	0	1	0	1	INIT[117]
1	1	1	0	1	1	0	INIT[118]
1	1	1	0	1	1	1	INIT[119]
1	1	1	1	0	0	0	INIT[120]
1	1	1	1	0	0	1	INIT[121]
1	1	1	1	0	1	0	INIT[122]
1	1	1	1	0	1	1	INIT[123]
1	1	1	1	1	0	0	INIT[124]
1	1	1	1	1	0	1	INIT[125]
1	1	1	1	1	1	0	INIT[126]
1	1	1	1	1	1	1	INIT[127]

4. Parameter description

Table 85 GTP_LUT7 parameter

Parameter Name	Describe	Defaults	Settings
INIT	Output configuration parameters	128'h00000000_00000000_00000000_00000000	0 ~ 2^128-1

5. Instantiation template

```
GTP_LUT7 #(
.INIT         (128'h00000000_00000000_00000000_00000000)
)GTP_LUT7_inst (

.Z            (z),
.I0           (i0),
.I1           (i1),
.I2           (i2),
.I3           (i3),
.I4           (i4),
.I5           (i5),
.I6           (i6)
);
```

(12) GTP_LUT8 instruction

1. Supported devices

Table 86 GTP_LUT8 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_LUT8 is an 8-input lookup table. Different INIT parameter values correspond to different functions of LUT8. When using it, you must specify the corresponding INIT parameter. Its structure diagram is as follows

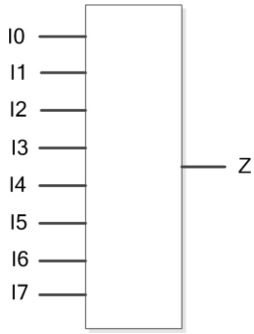


Figure 28 GTP_LUT8 structure diagram

3. Port description

Table 87 GTP_LUT8 port

Port Name	Direction	Functional description
Z	Output	Output signal
I0	Input	Input signal
I1	Input	Input signal
I2	Input	Input signal
I3	Input	Input signal
I4	Input	Input signal
I5	Input	Input signal
I6	Input	Input signal
I7	Input	Input signal

Truth table:

Table 88 GTP_LUT8 truth table

Input								Output
I7	I6	I5	I4	I3	I2	I1	I0	Z
0	0	0	0	0	0	0	0	INIT[0]
0	0	0	0	0	0	0	1	INIT[1]
0	0	0	0	0	0	1	0	INIT[2]
0	0	0	0	0	0	1	1	INIT[3]
0	0	0	0	0	1	0	0	INIT[4]
0	0	0	0	0	1	0	1	INIT[5]
0	0	0	0	0	1	1	0	INIT[6]
0	0	0	0	0	1	1	1	INIT[7]
0	0	0	0	1	0	0	0	INIT[8]
0	0	0	0	1	0	0	1	INIT[9]
0	0	0	0	1	0	1	0	INIT[10]
0	0	0	0	1	0	1	1	INIT[11]
0	0	0	0	1	1	0	0	INIT[12]

Input								Output
0	0	0	0	1	1	0	1	INIT[13]
0	0	0	0	1	1	1	0	INIT[14]
0	0	0	0	1	1	1	1	INIT[15]
0	0	0	1	0	0	0	0	INIT[16]
0	0	0	1	0	0	0	1	INIT[17]
0	0	0	1	0	0	1	0	INIT[18]
0	0	0	1	0	0	1	1	INIT[19]
0	0	0	1	0	1	0	0	INIT[20]
0	0	0	1	0	1	0	1	INIT[21]
0	0	0	1	0	1	1	0	INIT[22]
0	0	0	1	0	1	1	1	INIT[23]
0	0	0	1	1	0	0	0	INIT[24]
0	0	0	1	1	0	0	1	INIT[25]
0	0	0	1	1	0	1	0	INIT[26]
0	0	0	1	1	0	1	1	INIT[27]
0	0	0	1	1	1	0	0	INIT[28]
0	0	0	1	1	1	0	1	INIT[29]
0	0	0	1	1	1	1	0	INIT[30]
0	0	0	1	1	1	1	1	INIT[31]
0	0	1	0	0	0	0	0	INIT[32]
0	0	1	0	0	0	0	1	INIT[33]
0	0	1	0	0	0	1	0	INIT[34]
0	0	1	0	0	0	1	1	INIT[35]
0	0	1	0	0	1	0	0	INIT[36]
0	0	1	0	0	1	0	1	INIT[37]
0	0	1	0	0	1	1	0	INIT[38]
0	0	1	0	0	1	1	1	INIT[39]
0	0	1	0	1	0	0	0	INIT[40]
0	0	1	0	1	0	0	1	INIT[41]
0	0	1	0	1	0	1	0	INIT[42]
0	0	1	0	1	0	1	1	INIT[43]
0	0	1	0	1	1	0	0	INIT[44]
0	0	1	0	1	1	0	1	INIT[45]
0	0	1	0	1	1	1	0	INIT[46]
0	0	1	0	1	1	1	1	INIT[47]
0	0	1	1	0	0	0	0	INIT[48]
0	0	1	1	0	0	0	1	INIT[49]
0	0	1	1	0	0	1	0	INIT[50]
0	0	1	1	0	0	1	1	INIT[51]
0	0	1	1	0	1	0	0	INIT[52]
0	0	1	1	0	1	0	1	INIT[53]
0	0	1	1	0	1	1	0	INIT[54]
0	0	1	1	0	1	1	1	INIT[55]
0	0	1	1	1	0	0	0	INIT[56]
0	0	1	1	1	0	0	1	INIT[57]
0	0	1	1	1	0	1	0	INIT[58]

Input								Output
0	0	1	1	1	0	1	1	INIT[59]
0	0	1	1	1	1	0	0	INIT[60]
0	0	1	1	1	1	0	1	INIT[61]
0	0	1	1	1	1	1	0	INIT[62]
0	0	1	1	1	1	1	1	INIT[63]
0	1	0	0	0	0	0	0	INIT[64]
0	1	0	0	0	0	0	1	INIT[65]
0	1	0	0	0	0	1	0	INIT[66]
0	1	0	0	0	0	1	1	INIT[67]
0	1	0	0	0	1	0	0	INIT[68]
0	1	0	0	0	1	0	1	INIT[69]
0	1	0	0	0	1	1	0	INIT[70]
0	1	0	0	0	1	1	1	INIT[71]
0	1	0	0	1	0	0	0	INIT[72]
0	1	0	0	1	0	0	1	INIT[73]
0	1	0	0	1	0	1	0	INIT[74]
0	1	0	0	1	0	1	1	INIT[75]
0	1	0	0	1	1	0	0	INIT[76]
0	1	0	0	1	1	0	1	INIT[77]
0	1	0	0	1	1	1	0	INIT[78]
0	1	0	0	1	1	1	1	INIT[79]
0	1	0	1	0	0	0	0	INIT[80]
0	1	0	1	0	0	0	1	INIT[81]
0	1	0	1	0	0	1	0	INIT[82]
0	1	0	1	0	0	1	1	INIT[83]
0	1	0	1	0	1	0	0	INIT[84]
0	1	0	1	0	1	0	1	INIT[85]
0	1	0	1	0	1	1	0	INIT[86]
0	1	0	1	0	1	1	1	INIT[87]
0	1	0	1	1	0	0	0	INIT[88]
0	1	0	1	1	0	0	1	INIT[89]
0	1	0	1	1	0	1	0	INIT[90]
0	1	0	1	1	0	1	1	INIT[91]
0	1	0	1	1	1	0	0	INIT[92]
0	1	0	1	1	1	0	1	INIT[93]
0	1	0	1	1	1	1	0	INIT[94]
0	1	0	1	1	1	1	1	INIT[95]
0	1	1	0	0	0	0	0	INIT[96]
0	1	1	0	0	0	0	1	INIT[97]
0	1	1	0	0	0	1	0	INIT[98]
0	1	1	0	0	0	1	1	INIT[99]
0	1	1	0	0	1	0	0	INIT[100]
0	1	1	0	0	1	0	1	INIT[101]
0	1	1	0	0	1	1	0	INIT[102]
0	1	1	0	0	1	1	1	INIT[103]
0	1	1	0	1	0	0	0	INIT[104]

Input								Output
0	1	1	0	1	0	0	1	INIT[105]
0	1	1	0	1	0	1	0	INIT[106]
0	1	1	0	1	0	1	1	INIT[107]
0	1	1	0	1	1	0	0	INIT[108]
0	1	1	0	1	1	0	1	INIT[109]
0	1	1	0	1	1	1	0	INIT[110]
0	1	1	0	1	1	1	1	INIT[111]
0	1	1	1	0	0	0	0	INIT[112]
0	1	1	1	0	0	0	1	INIT[113]
0	1	1	1	0	0	1	0	INIT[114]
0	1	1	1	0	0	1	1	INIT[115]
0	1	1	1	0	1	0	0	INIT[116]
0	1	1	1	0	1	0	1	INIT[117]
0	1	1	1	0	1	1	0	INIT[118]
0	1	1	1	0	1	1	1	INIT[119]
0	1	1	1	1	0	0	0	INIT[120]
0	1	1	1	1	0	0	1	INIT[121]
0	1	1	1	1	0	1	0	INIT[122]
0	1	1	1	1	0	1	1	INIT[123]
0	1	1	1	1	1	0	0	INIT[124]
0	1	1	1	1	1	0	1	INIT[125]
0	1	1	1	1	1	1	0	INIT[126]
0	1	1	1	1	1	1	1	INIT[127]
1	0	0	0	0	0	0	0	INIT[128]
1	0	0	0	0	0	0	1	INIT[129]
1	0	0	0	0	0	1	0	INIT[130]
1	0	0	0	0	0	1	1	INIT[131]
1	0	0	0	0	1	0	0	INIT[132]
1	0	0	0	0	1	0	1	INIT[133]
1	0	0	0	0	1	1	0	INIT[134]
1	0	0	0	0	1	1	1	INIT[135]
1	0	0	0	1	0	0	0	INIT[136]
1	0	0	0	1	0	0	1	INIT[137]
1	0	0	0	1	0	1	0	INIT[138]
1	0	0	0	1	0	1	1	INIT[139]
1	0	0	0	1	1	0	0	INIT[140]
1	0	0	0	1	1	0	1	INIT[141]
1	0	0	0	1	1	1	0	INIT[142]
1	0	0	0	1	1	1	1	INIT[143]
1	0	0	1	0	0	0	0	INIT[144]
1	0	0	1	0	0	0	1	INIT[145]
1	0	0	1	0	0	1	0	INIT[146]
1	0	0	1	0	0	1	1	INIT[147]
1	0	0	1	0	1	0	0	INIT[148]
1	0	0	1	0	1	0	1	INIT[149]
1	0	0	1	0	1	1	0	INIT[150]

Input								Output
1	0	0	1	0	1	1	1	INIT[151]
1	0	0	1	1	0	0	0	INIT[152]
1	0	0	1	1	0	0	1	INIT[153]
1	0	0	1	1	0	1	0	INIT[154]
1	0	0	1	1	0	1	1	INIT[155]
1	0	0	1	1	1	0	0	INIT[156]
1	0	0	1	1	1	0	1	INIT[157]
1	0	0	1	1	1	1	0	INIT[158]
1	0	0	1	1	1	1	1	INIT[159]
1	0	1	0	0	0	0	0	INIT[160]
1	0	1	0	0	0	0	1	INIT[161]
1	0	1	0	0	0	1	0	INIT[162]
1	0	1	0	0	0	1	1	INIT[163]
1	0	1	0	0	1	0	0	INIT[164]
1	0	1	0	0	1	0	1	INIT[165]
1	0	1	0	0	1	1	0	INIT[166]
1	0	1	0	0	1	1	1	INIT[167]
1	0	1	0	1	0	0	0	INIT[168]
1	0	1	0	1	0	0	1	INIT[169]
1	0	1	0	1	0	1	0	INIT[170]
1	0	1	0	1	0	1	1	INIT[171]
1	0	1	0	1	1	0	0	INIT[172]
1	0	1	0	1	1	0	1	INIT[173]
1	0	1	0	1	1	1	0	INIT[174]
1	0	1	0	1	1	1	1	INIT[175]
1	0	1	1	0	0	0	0	INIT[176]
1	0	1	1	0	0	0	1	INIT[177]
1	0	1	1	0	0	1	0	INIT[178]
1	0	1	1	0	0	1	1	INIT[179]
1	0	1	1	0	1	0	0	INIT[180]
1	0	1	1	0	1	0	1	INIT[181]
1	0	1	1	0	1	1	0	INIT[182]
1	0	1	1	0	1	1	1	INIT[183]
1	0	1	1	1	0	0	0	INIT[184]
1	0	1	1	1	0	0	1	INIT[185]
1	0	1	1	1	0	1	0	INIT[186]
1	0	1	1	1	0	1	1	INIT[187]
1	0	1	1	1	1	0	0	INIT[188]
1	0	1	1	1	1	0	1	INIT[189]
1	0	1	1	1	1	1	0	INIT[190]
1	0	1	1	1	1	1	1	INIT[191]
1	1	0	0	0	0	0	0	INIT[192]
1	1	0	0	0	0	0	1	INIT[193]
1	1	0	0	0	0	1	0	INIT[194]
1	1	0	0	0	0	1	1	INIT[195]
1	1	0	0	0	1	0	0	INIT[196]

Input								Output
1	1	0	0	0	1	0	1	INIT[197]
1	1	0	0	0	1	1	0	INIT[198]
1	1	0	0	0	1	1	1	INIT[199]
1	1	0	0	1	0	0	0	INIT[200]
1	1	0	0	1	0	0	1	INIT[201]
1	1	0	0	1	0	1	0	INIT[202]
1	1	0	0	1	0	1	1	INIT[203]
1	1	0	0	1	1	0	0	INIT[204]
1	1	0	0	1	1	0	1	INIT[205]
1	1	0	0	1	1	1	0	INIT[206]
1	1	0	0	1	1	1	1	INIT[207]
1	1	0	1	0	0	0	0	INIT[208]
1	1	0	1	0	0	0	1	INIT[209]
1	1	0	1	0	0	1	0	INIT[210]
1	1	0	1	0	0	1	1	INIT[211]
1	1	0	1	0	1	0	0	INIT[212]
1	1	0	1	0	1	0	1	INIT[213]
1	1	0	1	0	1	1	0	INIT[214]
1	1	0	1	0	1	1	1	INIT[215]
1	1	0	1	1	0	0	0	INIT[216]
1	1	0	1	1	0	0	1	INIT[217]
1	1	0	1	1	0	1	0	INIT[218]
1	1	0	1	1	0	1	1	INIT[219]
1	1	0	1	1	1	0	0	INIT[220]
1	1	0	1	1	1	0	1	INIT[221]
1	1	0	1	1	1	1	0	INIT[222]
1	1	0	1	1	1	1	1	INIT[223]
1	1	1	0	0	0	0	0	INIT[224]
1	1	1	0	0	0	0	1	INIT[225]
1	1	1	0	0	0	1	0	INIT[226]
1	1	1	0	0	0	1	1	INIT[227]
1	1	1	0	0	1	0	0	INIT[228]
1	1	1	0	0	1	0	1	INIT[229]
1	1	1	0	0	1	1	0	INIT[230]
1	1	1	0	0	1	1	1	INIT[231]
1	1	1	0	1	0	0	0	INIT[232]
1	1	1	0	1	0	0	1	INIT[233]
1	1	1	0	1	0	1	0	INIT[234]
1	1	1	0	1	0	1	1	INIT[235]
1	1	1	0	1	1	0	0	INIT[236]
1	1	1	0	1	1	0	1	INIT[237]
1	1	1	0	1	1	1	0	INIT[238]
1	1	1	0	1	1	1	1	INIT[239]
1	1	1	1	0	0	0	0	INIT[240]
1	1	1	1	0	0	0	1	INIT[241]
1	1	1	1	0	0	1	0	INIT[242]

Input								Output
1	1	1	1	0	0	1	1	INIT[243]
1	1	1	1	0	1	0	0	INIT[244]
1	1	1	1	0	1	0	1	INIT[245]
1	1	1	1	0	1	1	0	INIT[246]
1	1	1	1	0	1	1	1	INIT[247]
1	1	1	1	1	0	0	0	INIT[248]
1	1	1	1	1	0	0	1	INIT[249]
1	1	1	1	1	0	1	0	INIT[250]
1	1	1	1	1	0	1	1	INIT[251]
1	1	1	1	1	1	0	0	INIT[252]
1	1	1	1	1	1	0	1	INIT[253]
1	1	1	1	1	1	1	0	INIT[254]
1	1	1	1	1	1	1	1	INIT[255]

4. Parameter description

Table 89 GTP_LUT8 parameter

Parameter Name	Describe	Defaults	Settings
INIT	output configuration setting parameters	256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000	0 ~ 2^256-1

5. Instantiation template

```
GTP_LUT8 #(
.INIT          (256'h0000000000000000_0000000000000000_0000000000000000_0000000000000000)
)GTP_LUT8_inst (

.Z             (z),
.I0            (i0),
.I1            (i1),
.I2            (i2),
.I3            (i3),
.I4            (i4),
.I5            (i5),
.I6            (i6),
.I7            (i7)
);
```

(13) GTP_LUT6CARRY instruction

1. Supported devices

Table 90 GTP_LUT6CARRY device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_LUT6CARRY is a fast carry logic, typical applications include addition and subtraction operations, counter counting that can be set, 4 to 1 data selector MUX41, numerical comparator, priority encoder, etc. Its structure diagram is as follows:

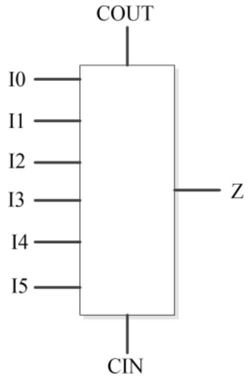


Figure 29 GTP_LUT6CARRY structure diagram

3. Port description

Table 91 GTP_LUT6CARRY port

Port Name	Direction	Functional description
I0	Input	Input signal
I1	Input	Input signal
I2	Input	Input signal
I3	Input	Input signal
I4	Input	Input signal
I5	Input	Input signal
CIN	Input	Input signal, cascade input
COUT	Output	Cascade output
Z	Output	Output signal

4. Parameter description

Table 92 GTP_LUT6CARRY parameter

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
INIT	<binary>	0~2^64-1	64'h0000_0000_0000_0000	Output configuration parameters
I5_TO_CARRY	<string>	"TRUE" or "FALSE"	"TRUE"	Select the parameters of I5, LUT5B
I5_TO_LUT	<string>	"TRUE" or "FALSE"	"FALSE"	Select the parameters of I5, CIN

5. Instantiation template

```
GTP_LUT6CARRY #(
.INIT(64'h0000_0000_0000_0000),
```

```
.I5_TO_CARRY("TRUE"),
.I5_TO_LUT("FALSE")
) GTP_LUT6CARRY_inst (
.COUT(cout),
.Z(z),
.CIN(cin),
.I0(i0),
.I1(i1),
.I2(i2),
.I3(i3),
.I4(i4),
.I5(i5)
);
```

(14) GTP_LUT6D instruction

1. Supported devices

Table 93 GTP_LUT6D device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_LUT6D is composed of a 6-input lookup table with 2 output ports. Different INIT parameter values correspond to different functions of LUT6D, and the corresponding INIT parameters must be specified when using. According to the input I5, I4, I3, I2, I1, I0 value, Z can output every bit of INIT, and Z5 can only output the lower 32 bits of INIT.

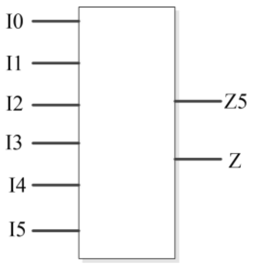


Figure 30 GTP_LUT6D structure diagram

3. Port description

Table 94 GTP_LUT6D port

Port Name	Direction	Functional description
I0	Input	Input signal
I1	Input	Input signal
I2	Input	Input signal
I3	Input	Input signal
I4	Input	Input signal

Port Name	Direction	Functional description
I5	Input	Input signal
Z5	Output	Output signal
Z	Output	Output signal

4. Parameter description

Table 95 GTP_LUT6D parameter

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
INIT	<binary>	0~2^64-1	64'h0000_0000_0000_0000	Output configuration parameters

Table 96 GTP_LUT6D parameter truth table

Input						Output	
I5	I4	I3	I2	I1	I0	Z	Z5
0	0	0	0	0	0	INIT[0]	INIT[0]
0	0	0	0	0	1	INIT[1]	INIT[1]
0	0	0	0	1	0	INIT[2]	INIT[2]
0	0	0	0	1	1	INIT[3]	INIT[3]
0	0	0	1	0	0	INIT[4]	INIT[4]
0	0	0	1	0	1	INIT[5]	INIT[5]
0	0	0	1	1	0	INIT[6]	INIT[6]
0	0	0	1	1	1	INIT[7]	INIT[7]
0	0	1	0	0	0	INIT[8]	INIT[8]
0	0	1	0	0	1	INIT[9]	INIT[9]
0	0	1	0	1	0	INIT[10]	INIT[10]
0	0	1	0	1	1	INIT[11]	INIT[11]
0	0	1	1	0	0	INIT[12]	INIT[12]
0	0	1	1	0	1	INIT[13]	INIT[13]
0	0	1	1	1	0	INIT[14]	INIT[14]
0	0	1	1	1	1	INIT[15]	INIT[15]
0	1	0	0	0	0	INIT[16]	INIT[16]
0	1	0	0	0	1	INIT[17]	INIT[17]
0	1	0	0	1	0	INIT[18]	INIT[18]
0	1	0	0	1	1	INIT[19]	INIT[19]
0	1	0	1	0	0	INIT[20]	INIT[20]
0	1	0	1	0	1	INIT[21]	INIT[21]
0	1	0	1	1	0	INIT[22]	INIT[22]
0	1	0	1	1	1	INIT[23]	INIT[23]
0	1	1	0	0	0	INIT[24]	INIT[24]
0	1	1	0	0	1	INIT[25]	INIT[25]
0	1	1	0	1	0	INIT[26]	INIT[26]
0	1	1	0	1	1	INIT[27]	INIT[27]
0	1	1	1	0	0	INIT[28]	INIT[28]
0	1	1	1	0	1	INIT[29]	INIT[29]
0	1	1	1	1	0	INIT[30]	INIT[30]
0	1	1	1	1	1	INIT[31]	INIT[31]
1	0	0	0	0	0	INIT[32]	INIT[0]

Input						Output	
1	0	0	0	0	1	INIT[33]	INIT[1]
1	0	0	0	1	0	INIT[34]	INIT[2]
1	0	0	0	1	1	INIT[35]	INIT[3]
1	0	0	1	0	0	INIT[36]	INIT[4]
1	0	0	1	0	1	INIT[37]	INIT[5]
1	0	0	1	1	0	INIT[38]	INIT[6]
1	0	0	1	1	1	INIT[39]	INIT[7]
1	0	1	0	0	0	INIT[40]	INIT[8]
1	0	1	0	0	1	INIT[41]	INIT[9]
1	0	1	0	1	0	INIT[42]	INIT[10]
1	0	1	0	1	1	INIT[43]	INIT[11]
1	0	1	1	0	0	INIT[44]	INIT[12]
1	0	1	1	0	1	INIT[45]	INIT[13]
1	0	1	1	1	0	INIT[46]	INIT[14]
1	0	1	1	1	1	INIT[47]	INIT[15]
1	1	0	0	0	0	INIT[48]	INIT[16]
1	1	0	0	0	1	INIT[49]	INIT[17]
1	1	0	0	1	0	INIT[50]	INIT[18]
1	1	0	0	1	1	INIT[51]	INIT[19]
1	1	0	1	0	0	INIT[52]	INIT[20]
1	1	0	1	0	1	INIT[53]	INIT[21]
1	1	0	1	1	0	INIT[54]	INIT[22]
1	1	0	1	1	1	INIT[55]	INIT[23]
1	1	1	0	0	0	INIT[56]	INIT[24]
1	1	1	0	0	1	INIT[57]	INIT[25]
1	1	1	0	1	0	INIT[58]	INIT[26]
1	1	1	0	1	1	INIT[59]	INIT[27]
1	1	1	1	0	0	INIT[60]	INIT[28]
1	1	1	1	0	1	INIT[61]	INIT[29]
1	1	1	1	1	0	INIT[62]	INIT[30]
1	1	1	1	1	1	INIT[63]	INIT[31]

5. Instantiation template

```
GTP_LUT6D #(
.INIT(64'h0000_0000_0000_0000)
) GTP_LUT6D_inst (
.Z(z),
.Z5(z5),
.I0(i0),
.I1(i1),
.I2(i2),
.I3(i3),
.I4(i4),
.I5(i5)
);
```

(15) GTP_MUX2LUT7 instruction

1. Supported devices

Table 97 GTP_MUX2LUT7 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_MUX2LUT7 is a dedicated two-input selector for GTP_LUT7, where I0 and I1 are data input, S is a selection signal, when S is 1'b0, it outputs I0, and when S is 1'b1, it outputs I1. Its structural block diagram is as follows. GTP_MUX2LUT7 function is to select the output of two LUT6s to form a LUT7, so it must be used with LUT6 at the same time.

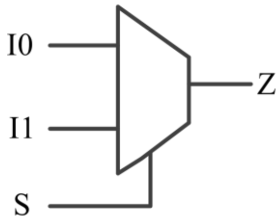


Figure 31 GTP_MUX2LUT7 structure diagram

3. Port description

Table 98 GTP_MUX2LUT7 port

Port Name	Direction	Functional description
I0	Input	Input signal
I1	Input	Input signal
S	Input	Select signal
Z	Output	Output signal

4. Instantiation template

```
GTP_MUX2LUT7 GTP_MUX2LUT7_inst (
.Z(lut7_out),
.I0(lut6a_out),
.I1(lut6b_out),
.S(i6)
);
```

(16) GTP_MUX2LUT8 instruction

1. Supported devices

Table 99 GTP_MUX2LUT8 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_MUX2LUT8 is a dedicated two-input selector for GTP_LUT8, where I0 and I1 are data inputs, S is a selection signal, when S is 1'b0, it outputs I0, and when S is 1'b1, it outputs I1. Its structural block diagram is shown below. GTP_MUX2LUT8 is to select the output of two LUT7 to form a LUT8, so it must be used with LUT7 at the same time.

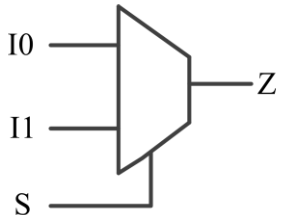


Figure 32 GTP_MUX2LUT8 structure diagram

3. Port description

Table 100 GTP_MUX2LUT8 port

Port Name	Direction	Functional description
I0	Input	Input signal
I1	Input	Input signal
S	Input	Select signal
Z	Output	Output signal

4. Instantiation template

```
GTP_MUX2LUT8 GTP_MUX2LUT8_inst (
.Z(lut8_out),
.I0(lut7ab_out),
.I1(lut7cd_out),
.S(i7)
);
```

4. ROM and RAM related GTP instructions

(1) General introduction

ROM&RAM related GTP mainly implements various registers.The GTP_RAM family implements single/dual input port simple registers. GTP_DRM series implements dual-port/simple dual-port/single-port registers, and implements three different write modes, NW, TW and RBW modes. GTP_FIFO implements the register that prompts the empty and full state, and has the function of packet loss and retransmission.

When distributed RAM or DRM reads and writes data through two ports (dual port, simple dual port mode), two ports cannot read and write the same address at the same time.

(2) GTP_ROM32X1 instruction

1. Supported devices

Table 101 GTP_ROM32X1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_ROM32X1 is a storage ROM with an address depth of 32 and a data bit width of 1 bit. The block diagram is as follows.

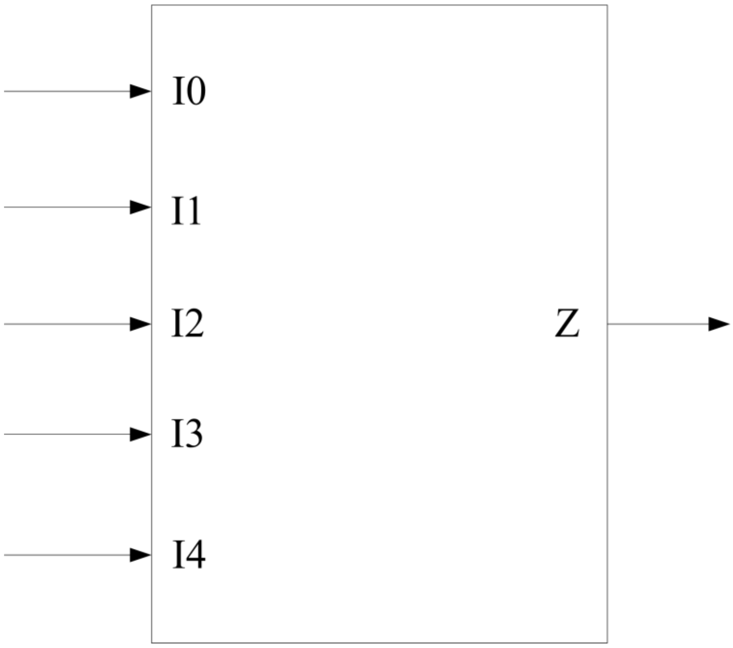


Figure 33 GTP_ROM32X1 structure diagram

3. Port description

Table 102 GTP_ROM32X1 port list

Port Name	Direction	Functional description
I0	Input	ROM read address addr[0]
I1	Input	ROM read address addr[1]

Port Name	Direction	Functional description
I2	Input	ROM read address addr[2]
I3	Input	ROM read address addr[3]
I4	Input	ROM read address addr[4]
Z	Output	Read data

4. Parameter description

Table 103 GTP_ROM32X1 parameter list

Parameter Name	Parameter Type	Settings	Defaults	Describe
INIT	<binary>	32'h00000000~32'hffffffff	32'h00000000	ROM initialization configuration parameters

5. Instantiation template

```
GTP_ROM32X1 #(
.INIT         (32'h00000000)
)
GTP_ROM32X1_inst(
.I0           (i0),
.I1           (ii),
.I2           (i2),
.I3           (i3),
.I4           (i4),
.Z            (z )
);
```

6. Detailed function description

The GTP implements the storage ROM function. Input I4~I0 to form the read data address, and read the value of the specified bit of the ROM initial configuration parameter.

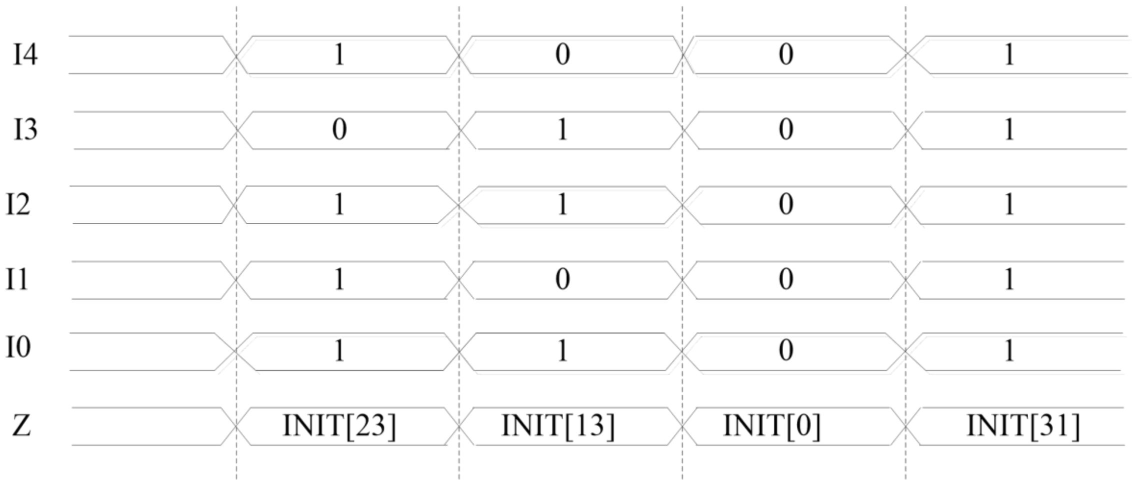


Figure 34 GTP_ROM32X1 waveform diagram

(3) GTP_ROM32X2 instruction

1. Supported devices

Table 104 GTP_ROM32X2 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_ROM32X2 is a storage ROM with an address depth of 32 and a data bit width of 2 bits. The structure diagram is as follows:

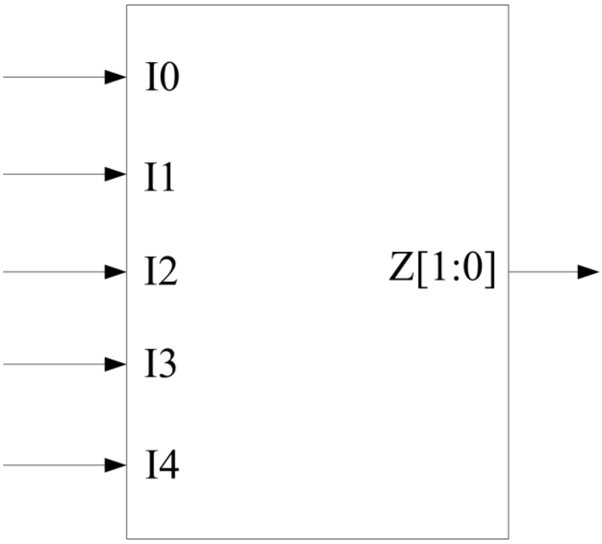


Figure 35 GTP_ROM32X2 structure diagram

3. Port description

Table 105 GTP_ROM32X2 port list

Port Name	Direction	Functional description
I0	Input	ROM read address addr[0]
I1	Input	ROM read address addr[1]
I2	Input	ROM read address addr[2]
I3	Input	ROM read address addr[3]
I4	Input	ROM read address addr[4]
Z	Output	Read data

4. Parameter description

Table 106 GTP_ROM32X2 parameter list

Parameter Name	Parameter Type	Settings	Defaults	Describe
INIT_0	<binary>	32'h0~32'hffffffff	32'h0	ROM initialization configuration parameters
INIT_1	<binary>	32'h0~32'hffffffff	32'h0	ROM initialization configuration parameters

5. Instantiation template

```
GTP_ROM32X2 #(
```

```
.INIT_0       (32'h00000000),
.INIT_1       (32'h00000000)
)
GTP_ROM32X2_inst(

.I0 (i0),
.I1 (ii),
.I2 (i2),
.I3 (i3),
.I4 (i4),
.Z  (z )
);
```

6. Detailed function description

The GTP implements the storage ROM function. Input I4~I0 to form the read data address, and read the value of the specified bit of the ROM initial configuration parameter.

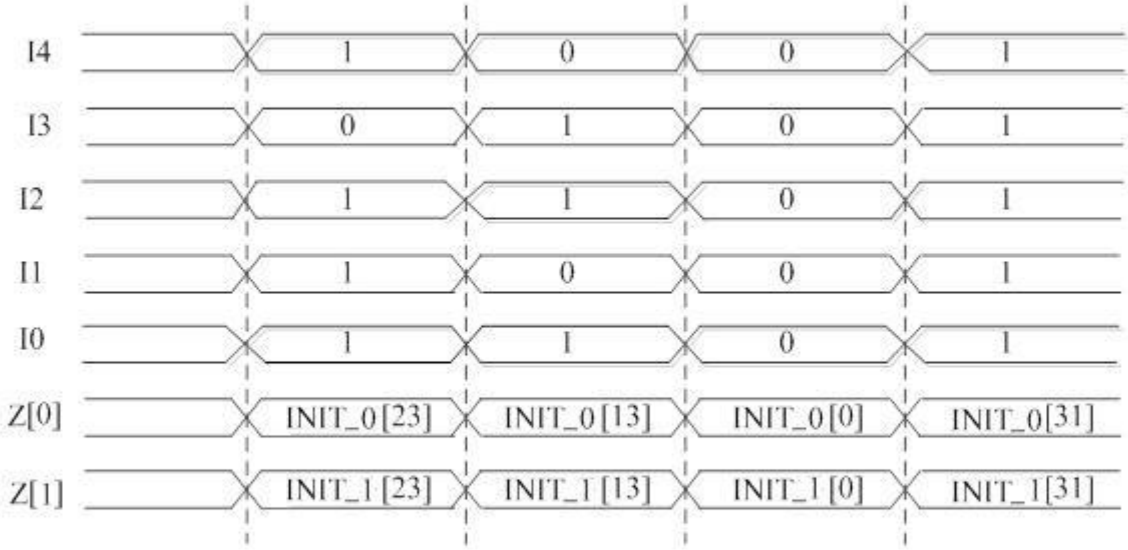


Figure 36 GTP_ROM32X2 waveform diagram

(4) GTP_ROM64X1 instruction

1. Supported devices

Table 107 GTP_ROM64X1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_ROM64X1 is a storage ROM with an address depth of 64 and a data bit width of 1 bit. The block diagram is as follows.

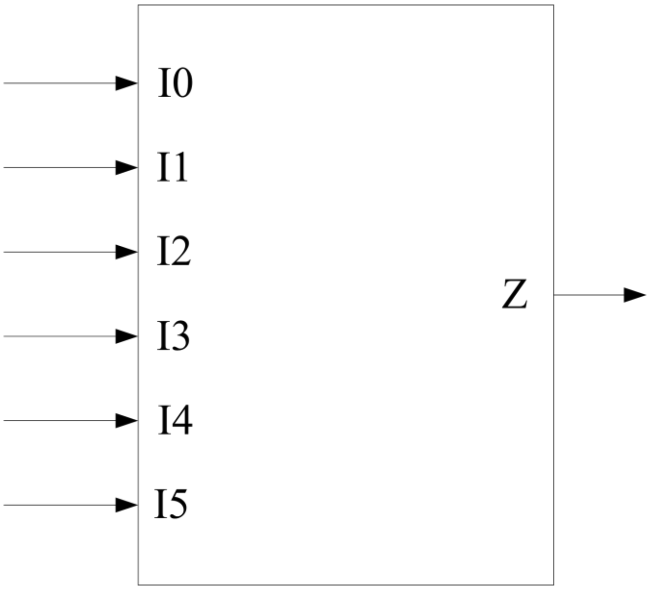


Figure 37 GTP_ROM64X1 structure diagram

3. Port description

Table 108 GTP_ROM64X1 port list

Port Name	Direction	Functional description
I0	Input	ROM read address addr[0]
I1	Input	ROM read address addr[1]
I2	Input	ROM read address addr[2]
I3	Input	ROM read address addr[3]
I4	Input	ROM read address addr[4]
I5	Input	ROM read address addr[5]
Z	Output	Read data

4. Parameter description

Table 109 GTP_ROM64X1 parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT	<binary>	64'h0~64'hffffffff_ffffffff	64'h0	ROM initialization configuration parameters

5. Instantiation template

```
GTP_ROM64X1#(
```

```
.INIT         (64'h00000000_00000000)
)
GTP_ROM64X1_inst(
.I0 (i0),
.I1 (ii),
.I2 (i2),
.I3 (i3),
.I4 (i4),
.I5 (i5),
.Z  (z )
);
```

6. Detailed function description

The GTP implements the function of storing ROM. Input I5~I0 to form the read data address, and read the value of the specified bit of the ROM initial configuration parameter.

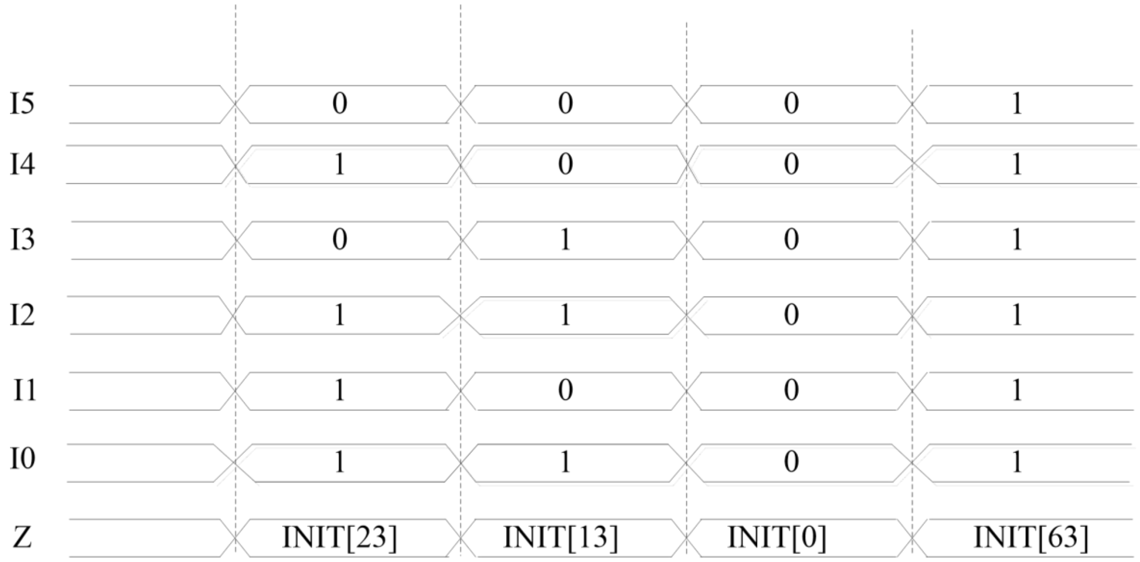


Figure 38 GTP_ROM64X1 waveform diagram

(5) GTP_ROM128X1 instruction

1. Supported devices

Table 110 GTP_ROM128X1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_ROM128X1 is a storage ROM with an address depth of 128 and a data bit width of 1 bit. The block diagram is as follows.

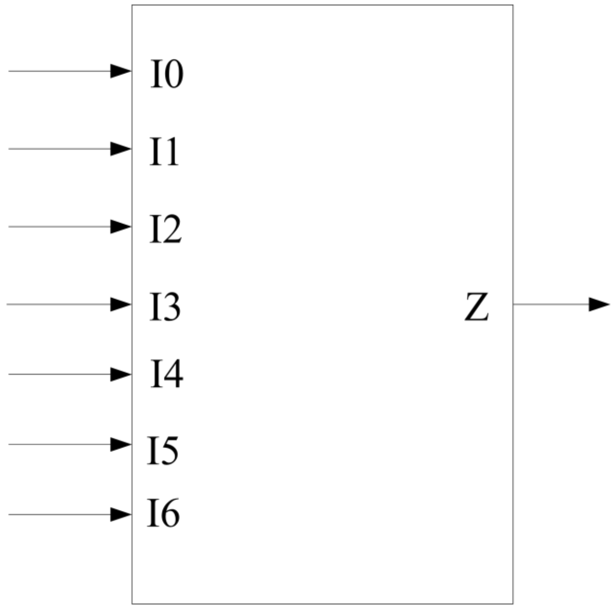


Figure 39 GTP_ROM128X1 structure diagram

3. Port description

Table 111 GTP_ROM128X1 port list

Port Name	Direction	Functional description
I0	Input	ROM read address addr[0]
I1	Input	ROM read address addr[1]
I2	Input	ROM read address addr[2]
I3	Input	ROM read address addr[3]
I4	Input	ROM read address addr[4]
I5	Input	ROM read address addr[5]
I6	Input	ROM read address addr[6]
Z	Output	Read data

4. Parameter description

Table 112 GTP_ROM128X1 parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT	<binary>	128'h00000000_00000000_00000000_00000000~128'hffffffff_ffffffff_ffffffff_ffffffff	128'h00000000_00000000_00000000_00000000	ROM initialization configuration parameters

5. Instantiation template

```
GTP_ROM128X1  #(
.INIT        (128'h00000000_00000000_00000000_00000000)
)
GTP_ROM128X1_inst(
.I0 (i0),
.I1 (ii),
.I2 (i2),
.I3 (i3),
```

```
.I4 (i4),
.I5 (i5),
.I6 (i6),
.Z  (z )
);
```

6. Detailed function description

The GTP implements the function of storing ROM. Input I6~I0 to form the read data address, and read the value of the specified bit of the ROM initial configuration parameter.

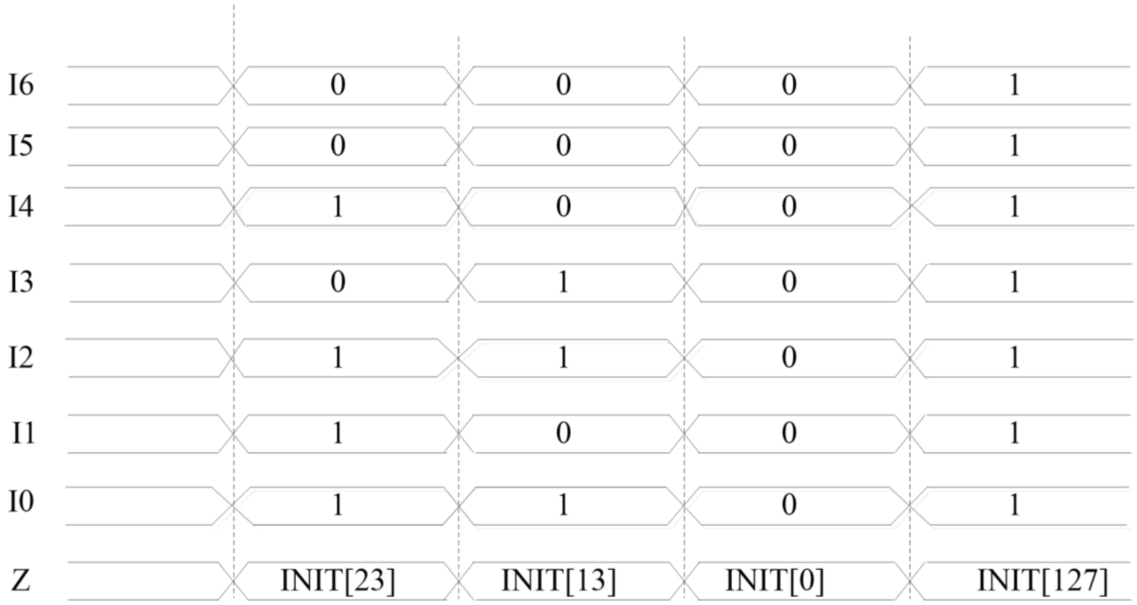


Figure 40 GTP_ROM128X1 waveform diagram

(6) GTP_ROM256X1 instruction

1. Supported devices

Table 113 GTP_ROM256X1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_ROM256X1 is a storage ROM with an address depth of 256 and a data bit width of 1 bit. The block diagram is as follows.

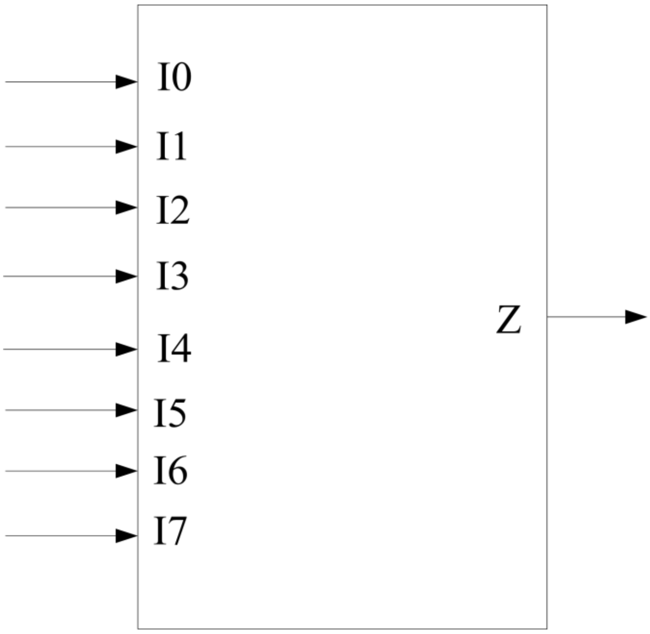


Figure 41 GTP_ROM256X1 structure diagram

3. Port description

Table 114 GTP_ROM256X1 port list

Port Name	Direction	Functional description
I0	Input	ROM read address addr[0]
I1	Input	ROM read address addr[1]
I2	Input	ROM read address addr[2]
I3	Input	ROM read address addr[3]
I4	Input	ROM read address addr[4]
I5	Input	ROM read address addr[5]
I6	Input	ROM read address addr[6]
I7	Input	ROM read address addr[7]
Z	Output	Read data

4. Parameter description

Table 115 GTP_ROM256X1 parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT	<binary>	256'h0~256'hffffffff_ffffffff_ffffffff_ffff ffff_ffffffff_ffffffff_ffffffff_ffffffff	256'h0	ROM initialization configuration setting parameters

5. Instantiation template

```
GTP_ROM256X1  #(
.INIT         (256'h0)
)
GTP_ROM256X1_inst(
.I0           (i0),
.I1           (ii),
.I2           (i2),
.I3           (i3),
```

```
.I4            (i4),
.I5            (i5),
.I6            (i6),
.I7            (i7),
.Z             (z )
);
```

6. Detailed function description

The GTP realizes the function of storing ROM. Input I7~I0 to form the read data address, and read the value of the specified bit of the ROM initial configuration parameter.

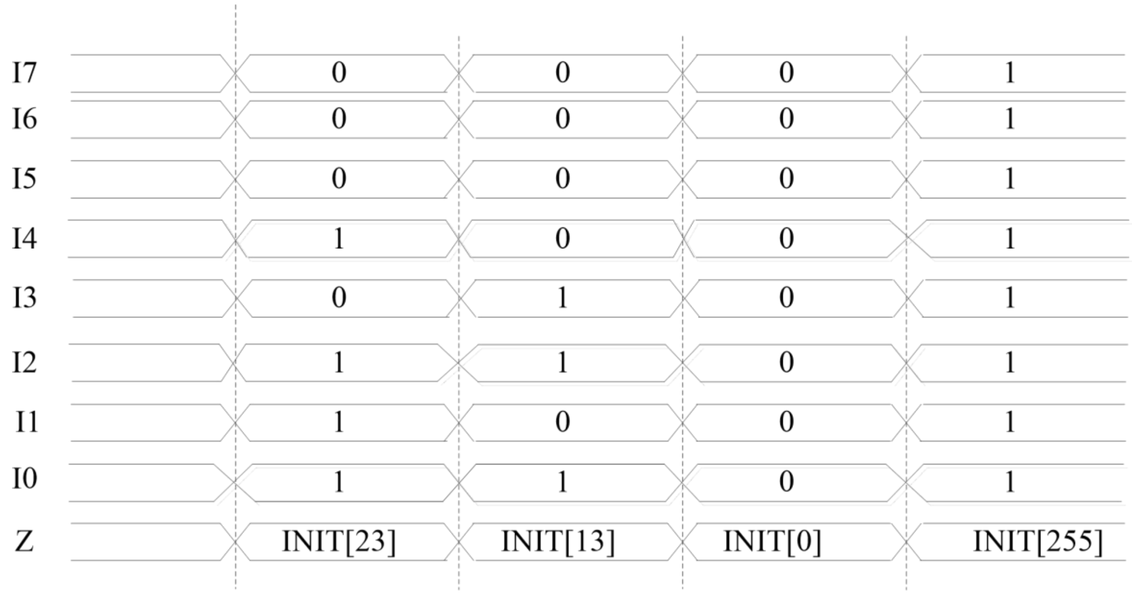


Figure 42 GTP_ROM256X1 waveform diagram

(7) GTP_RAM32X1SP instruction

1. Supported devices

Table 116 GTP_RAM32X1SP device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_RAM32X1SP is a single-port random memory module with a data bit width of 1 and an address depth of 32. The block diagram is as follows.

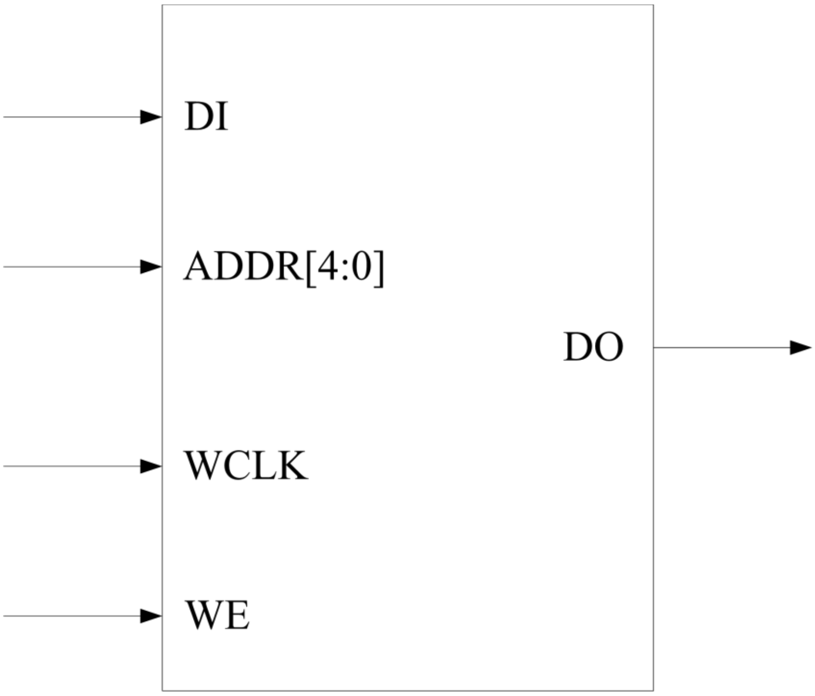


Figure 43 GTP_RAM32X1SP structure diagram

3. Port description

Table 117 GTP_RAM32X1SP port list

Port Name	Direction	Functional description
DI	Input	Data input port
ADDR	Input	Read and write address
WCLK	Input	Write clock
WE	Input	Write enable
DO	Output	Data output port

4. Parameter description

Table 118 GTP_RAM32X1SP parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT	<binary>	32'h0~32'hffffffff	32'h0	Memory initialization configuration parameters

5. Instantiation template

```
GTP_RAM32X1SP#(
.INIT          (32'h00000000)
)
GTP_RAM32X1SP_inst(
.DO            (do       ),
.DI            (di       ),
.ADDR          (addr     ),
.WCLK          (wclk     ),
.WE            (we       )
);
```

6. Detailed function description

The GTP implements the single-port RAM function, and only performs write operations when WE is high. When WE is high, the input data DI is written into the current address ADDR at the rising edge of WCLK, and the input data is read from the DO port. Output. When WE is low level, the data in the current address ADDR is read from the DO port at the rising edge of WCLK, and no write operation is performed. As shown in the figure below.

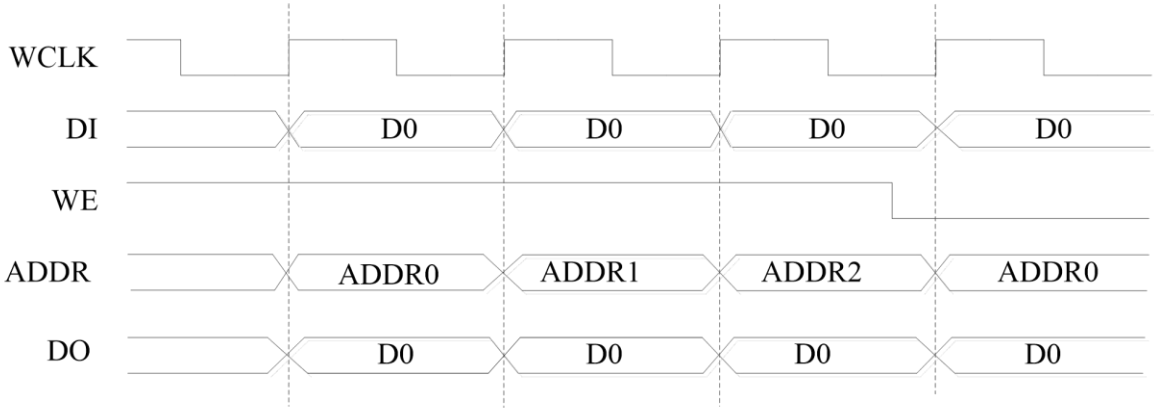


Figure 44 GTP_RAM32X1SP waveform diagram

(8) GTP_RAM32X1DP instruction

1. Supported devices

Table 119 GTP_RAM32X1DP device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_RAM32X1DP is a dual-port random memory module with a data bit width of 1 and an address depth of 32. The read and write addresses are input by two independent ports. The block diagram is as follows.

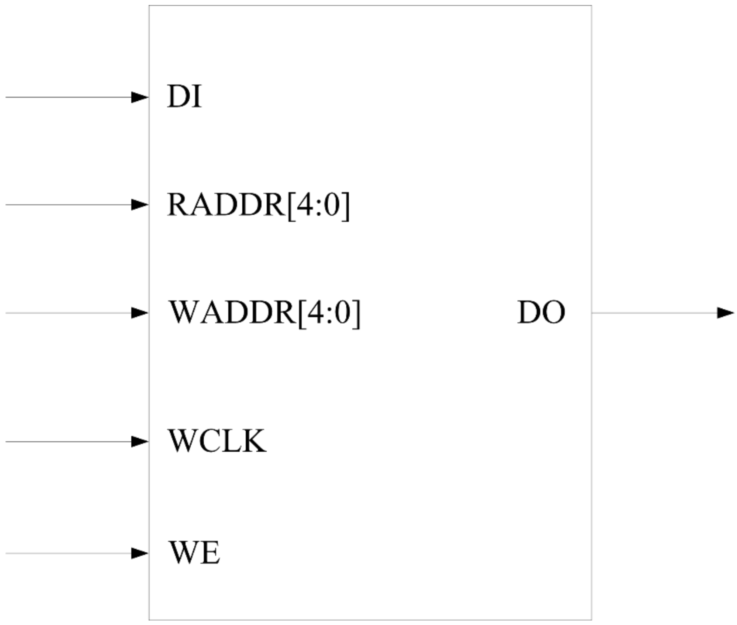


Figure 45 GTP_RAM32X1DP structure diagram

3. Port description

Table 120 GTP_RAM32X1DP port list

Port Name	Direction	Functional description
DI	Input	Data input port
RADDR	Input	Read address
WADDR	Input	Write address
WCLK	Input	Write clock
WE	Input	Write enable
DO	Output	Data output port

4. Parameter description

Table 121 GTP_RAM32X1DP parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT	<binary>	32'h0~32'hffffffff	32'h0	Memory initialization configuration parameters

5. Instantiation template

```
GTP_RAM32X1DP #(
.INIT         (32'h0 )
)
GTP_RAM32X1DP_inst(
.DO           (do       ),
.DI           (di       ),
.RADD         (raddr    ),
.WADDR        (waddr    ),
.WCLK         (wclk     ),
.WE           (we       )
);
```

6. Detailed function description

The GTP realizes the function of dual-port RAM, and the read and write addresses are independent of each other. Only when WE is high level, the input data is write into RAM. When WE is high level, the input data DI is written into the write address WADDR at the rising edge of WCLK In, the data in the read address RADDR is read from the DO port. When WE is low, the data in the read address RADDR is read from the DO port at the rising edge of WCLK, and no write operation is performed.

Reading and writing are performed asynchronously, and the read and write addresses cannot be the same at the same time to prevent read and write conflicts. As shown in the figure below.

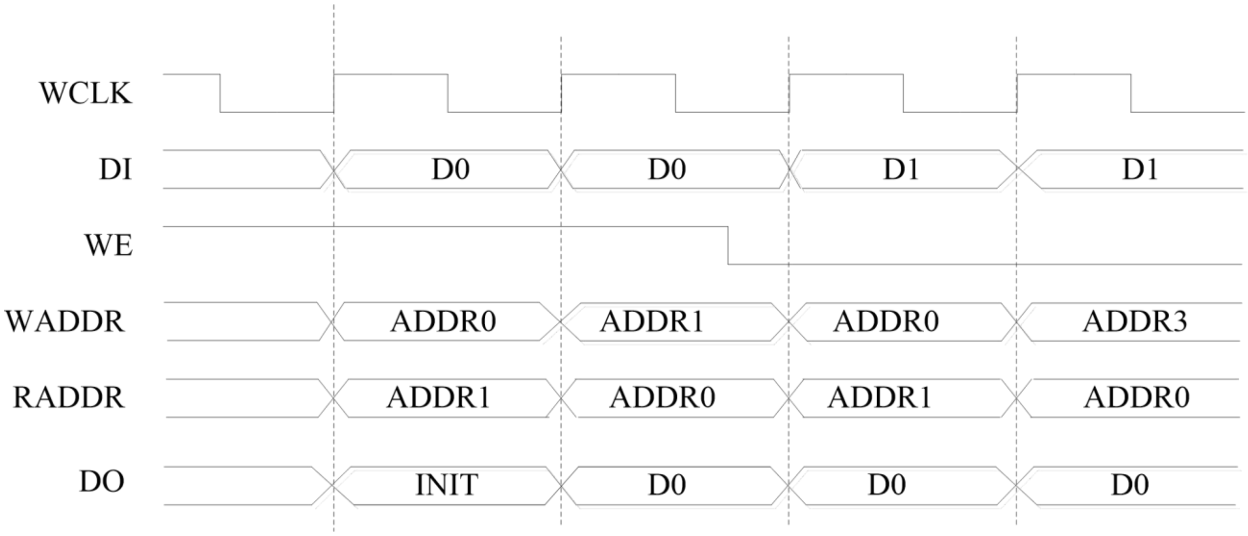


Figure 46 GTP_RAM32X1DP waveform diagram

(9) GTP_RAM32X2SP instruction

1. Supported devices

Table 122 GTP_RAM32X2SP device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_RAM32X2SP is a single-port random memory module with a data bit width of 2 and an address depth of 32. The block diagram is as follows.

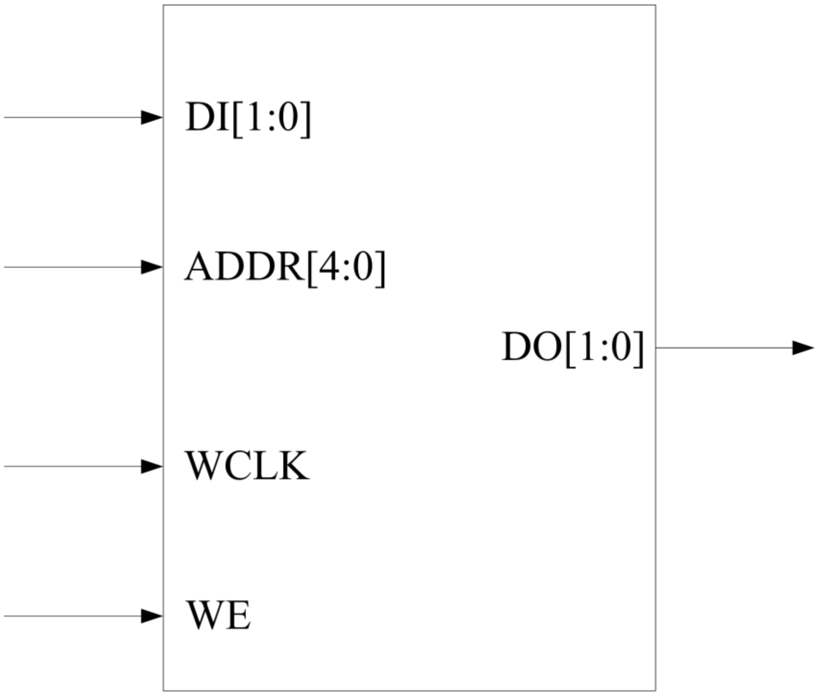


Figure 47 GTP_RAM32X2SP structure diagram

3. Port description

Table 123 GTP_RAM32X2SP port list

Port Name	Direction	Functional description
DI[1:0]	Input	Data input port
ADDR	Input	Read and write address
WCLK	Input	Write clock
WE	Input	Write enable
DO[1:0]	Output	Data output port

4. Parameter description

Table 124 GTP_RAM32X2SP parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT_0	<binary>	32'h0~32'hffffffff	32'h0	Memory initialization configuration parameters
INIT_1	<binary>	32'h0~32'hffffffff	32'h0	Memory initialization configuration parameters

5. Instantiation template

```
GTP_RAM32X2SP#(
.INIT_0       (32'h0),
.INIT_1       (32'h0)
)
GTP_RAM32X2SP_inst(
.DO           (do       ),
.DI           (di       ),
.ADDR         (addr     ),
.WCLK         (wclk     ),
.WE           (we       )
);
```

(10) GTP_RAM32X2DP instruction

1. Supported devices

Table 125 GTP_RAM32X2DP device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_RAM32X2DP is a dual-port random memory module with a data bit width of 2 and an address depth of 32. The read and write addresses are input by two independent ports. The block diagram is as follows.

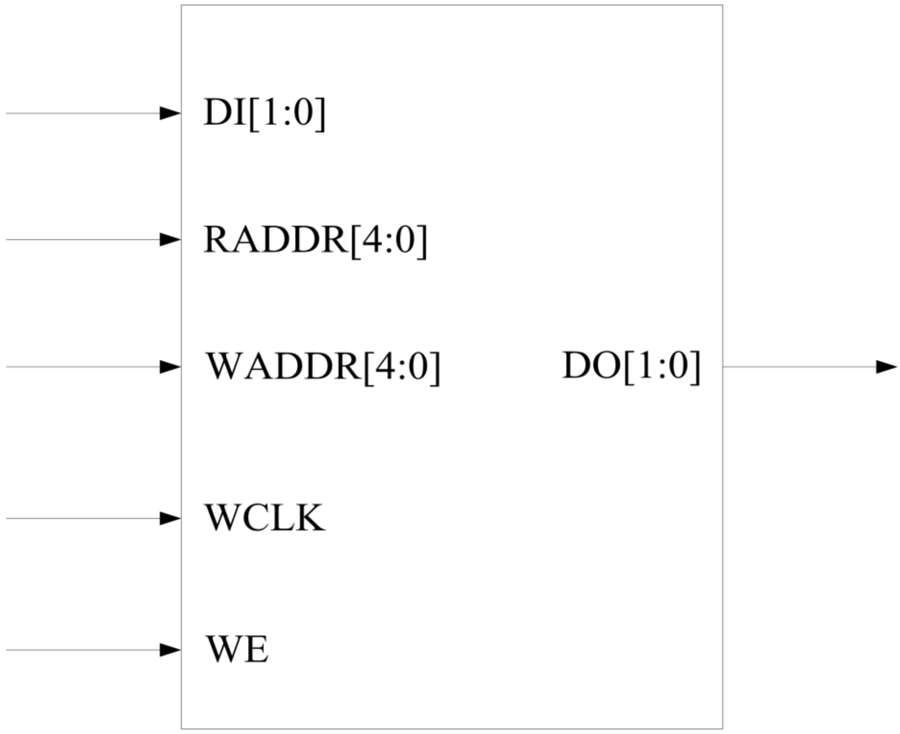


Figure 48 GTP_RAM32X2DP structure diagram

3. Port description

Table 126 GTP_RAM32X2DP port list

Port Name	Direction	Functional description
DI	Input	Data input port
RADDR	Input	Read address
WADDR	Input	Write address
WCLK	Input	Write clock
WE	Input	Write enable
DO	Output	Data output port

4. Parameter description

Table 127 GTP_RAM32X2DP parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT_0	<binary>	32'h0~32'hffffffff	32'h0	Memory initialization configuration parameters
INIT_1	<binary>	32'h0~32'hffffffff	32'h0	Memory initialization configuration parameters

5. Instantiation template

```
GTP_RAM32X2DP #(
.INIT_0       (32'h0 ),
.INIT_1       (32'h0 )
)
GTP_RAM32X2DP_inst(
.DO           (do       ),
.DI           (di       ),
.RADDR        (raddr    ),
```

```
.WADDR          (waddr    ),
.WCLK           (wclk     ),
.WE             (we       )
);
```

(11) GTP_RAM64X1SP instruction

1. Supported devices

Table 128 GTP_RAM64X1SP device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_RAM64X1SP is a dual-port random memory module with a data bit width of 1 and an address depth of 64. The read and write addresses are input by two independent ports. The block diagram is as follows.

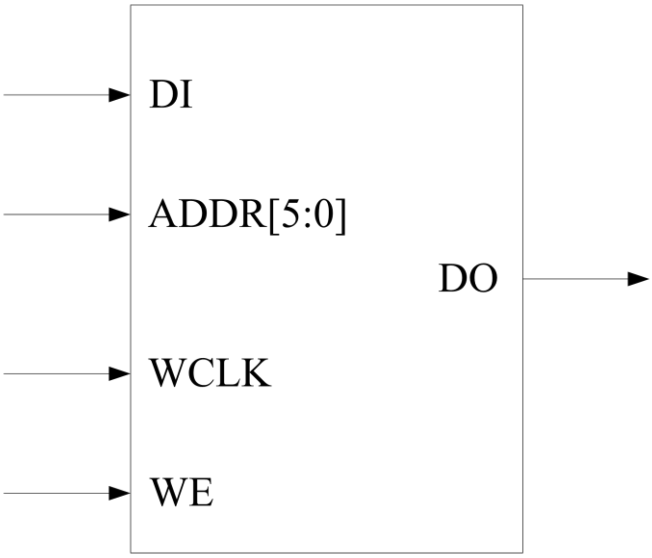


Figure 49 GTP_RAM64X1SP structure diagram

3. Port description

Table 129 GTP_RAM64X1SP port list

Port Name	Direction	Functional description
DI	Input	Write data
ADDR	Input	Write address
WCLK	Input	Write clock
WE	Input	Write enable
DO	Output	Read data

4. Parameter description

Table 130 GTP_RAM64X1SP parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT	<binary>	64'h0~64'hffffffffffffffff	64'h0	Memory initialization configuration parameters

5. Instantiation template

```
GTP_RAM64X1SP#(
.INIT          (64'h0 )
)
GTP_RAM64X1SP_inst(
.DO            (do       ),
.DI            (di       ),
.ADDR          (addr     ),
.WCLK          (wclk     ),
.WE            (we       )
);
```

6. Detailed function description

The GTP realizes the single-port RAM function, and only writes when WE is high. When WE is high, the input data DI is written into the current address ADDR at the rising edge of WCLK, and the input data is read from the DO port. Output. When WE is low level, the data in the current address ADDR is read from the DO port at the rising edge of WCLK, and no write operation is performed. As shown in the figure below.

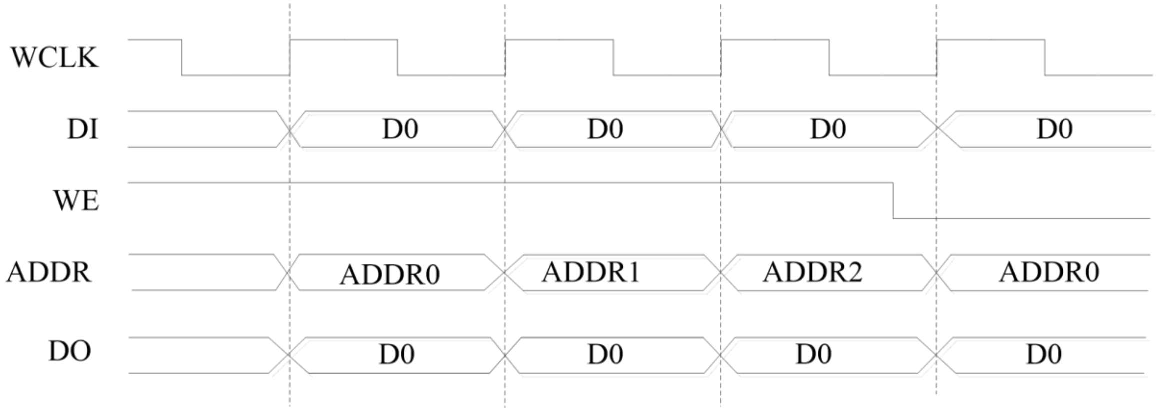


Figure 50 GTP_RAM64X1SP waveform diagram

(12) GTP_RAM64X1DP instruction

1. Supported devices

Table 131 GTP_RAM64X1DP device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_RAM64X1DP is a dual-port random memory module with a data bit width of 1 and an address depth of 64. The read and write addresses are input by two independent ports. The block diagram is as follows.

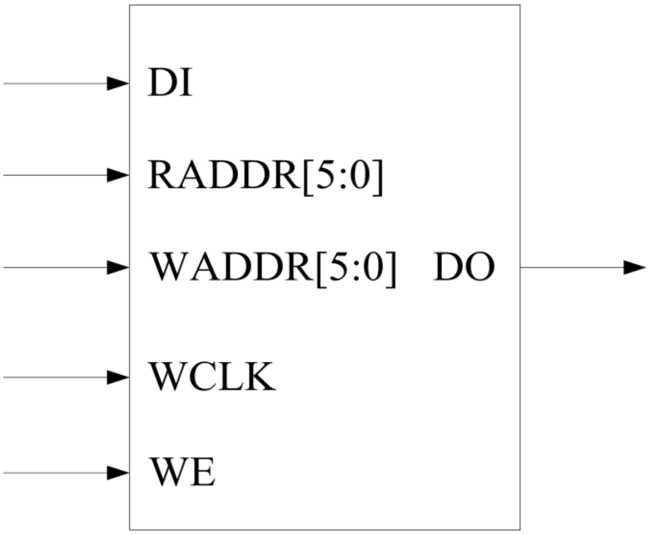


Figure 51 GTP_RAM64X1DP structure diagram

3. Port description

Table 132 GTP_RAM64X1DP port list

Port Name	Direction	Functional description
DI	Input	Write data
RADDR	Input	Read address
WADDR	Input	Write address
WCLK	Input	Write clock
WE	Input	Write enable
DO	Output	Read data

4. Parameter description

Table 133 GTP_RAM64X1DP parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT	<binary>	64'h0~64'hffffffffffffffff	64'h00000000	Memory initialization configuration parameters

5. Instantiation template

```
GTP_RAM64X1DP #(
.INIT         (64'h0)
)
GTP_RAM64X1DP_inst(
.DO           (do        ),
.DI           (di        ),
.RADDR        (raddr     ),
.WADDR        (waddr     ),
.WCLK         (wclk      ),
.WE           (we        )
);
```

6. Detailed function description

The GTP realizes the dual-port RAM function, and the read and write addresses are independent of each other. Only when WE is high, the input data will be write into RAM. When WE is high level, the input data DI will be written into the

write address WADDR at the rising edge of WCLK, and the data in the read address RADDR will be read out from the DO port. When WE is low level, it will be read at the rising edge of WCLK The data in the address RADDR is read from the DO port, not written.

Reading and writing are performed asynchronously, and the read and write addresses cannot be the same at the same time to prevent read and write conflicts. As shown in the figure below.

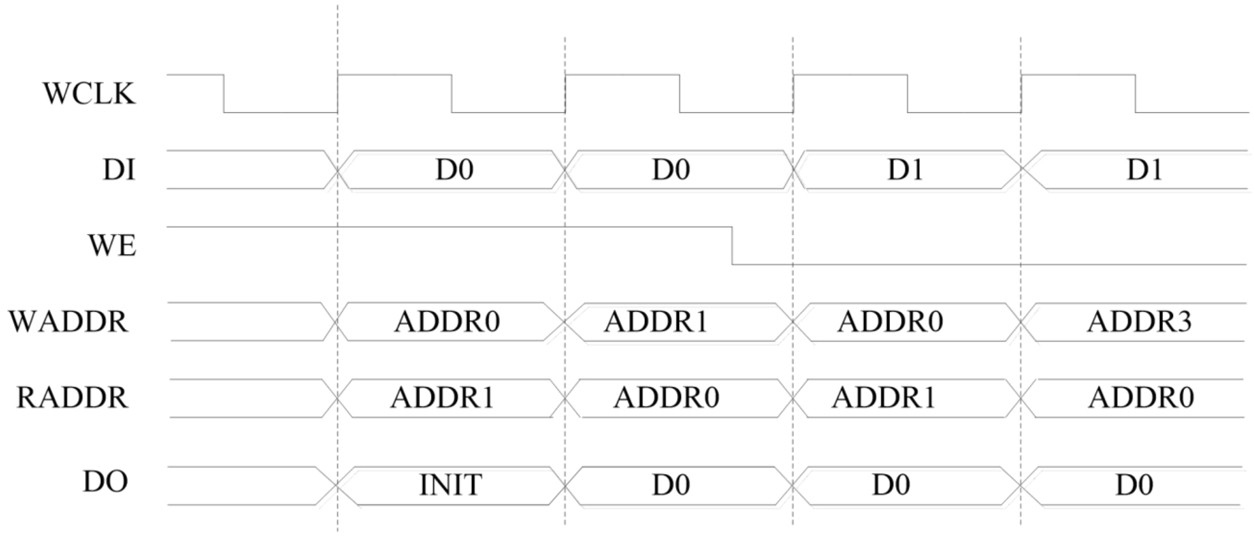


Figure 52 GTP_RAM64X1DP waveform diagram

(13) GTP_RAM128X1SP instruction

1. Supported devices

Table 134 GTP_RAM128X1SP device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_RAM128X1SP is a dual-port random memory module with a data bit width of 1 and an address depth of 128. The read and write addresses are input by two independent ports. The block diagram is as follows.

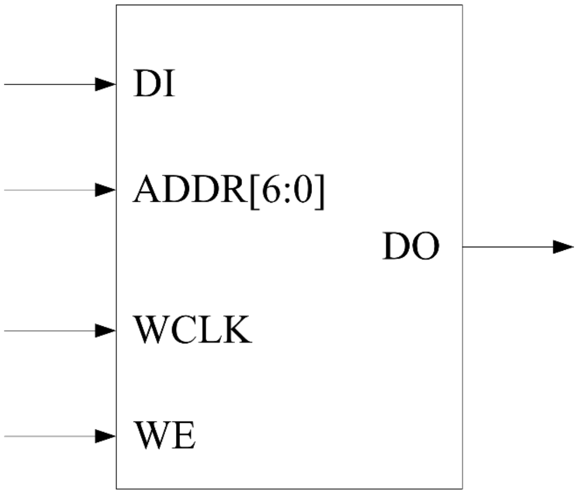


Figure 53 GTP_RAM128X1SP structure diagram

3. Port description

Table 135 GTP_RAM128X1SP port list

Port Name	Direction	Functional description
DI	Input	Write data
ADDR	Input	Write address
WCLK	Input	Write clock
WE	Input	Write enable
DO	Output	Read data

4. Parameter description

Table 136 GTP_RAM128X1SP parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT	<binary>	128'h0~128'hffffffffffffffffffffffffffffffff	128'h0	Memory initialization configuration parameters

5. Instantiation template

```
GTP_RAM128X1SP#(
.INIT         (128'h0)
) GTP_RAM128X1SP_inst(
.DO           (do        ),
.DI           (di        ),
.ADDR         (addr      ),
.WCLK         (wclk      ),
.WE           (we        )
);
```

6. Detailed function description

The GTP realizes the single-port RAM function, and only writes when WE is high. When WE is high, the input data DI is written into the current address ADDR at the rising edge of WCLK, and the input data is read from the DO port. Output. When WE is low level, the data in the current address ADDR is read from the DO port at the rising edge of WCLK, and no write operation is performed. As shown in the figure below.

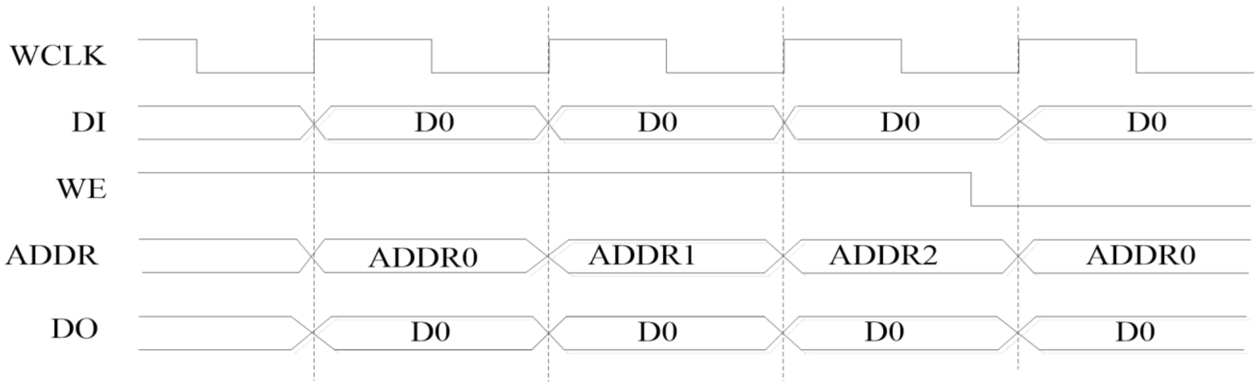


Figure 54 GTP_RAM128X1SP waveform diagram

(14) GTP_RAM128X1DP instruction

1. Supported devices

Table 137 GTP_RAM128X1DP device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_RAM128X1DP is a dual-port random memory module with a data bit width of 1 and an address depth of 128. The read and write addresses are input by two independent ports. The block diagram is as follows.

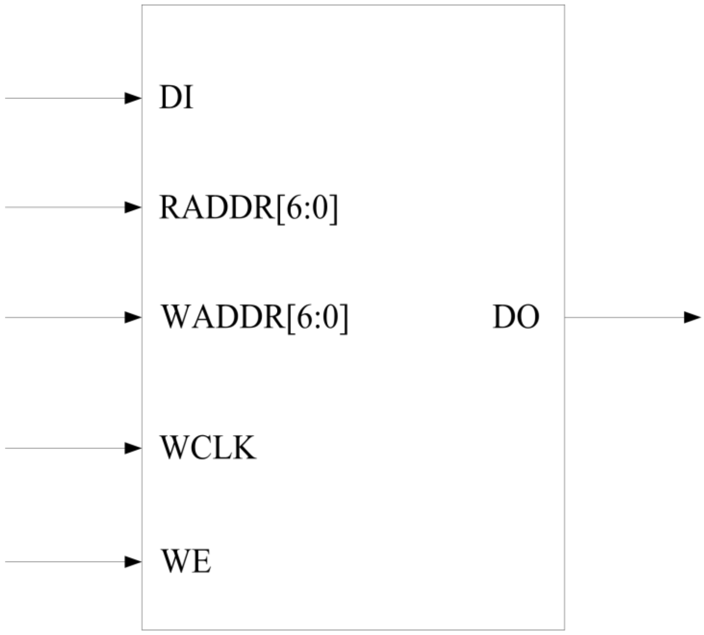


Figure 55 GTP_RAM128X1DP structure diagram

3. Port description

Table 138 GTP_RAM128X1DP port list

Port Name	Direction	Functional description
DI	Input	Write data
RADDR	Input	Read address
WADDR	Input	Write address
WCLK	Input	Write clock
WE	Input	Write enable
DO	Output	Read data

4. Parameter description

Table 139 GTP_RAM128X1DP parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT	<binary>	128'h0~128'hffffffffffffffffffffffffffffffff	128'h0	Memory initialization configuration parameters

5. Instantiation template

```
GTP_RAM128X1DP #(
.INIT          (128'h0)
) GTP_RAM128X1DP_inst(
.DO            (do       ),
.DI            (di       ),
.RADDR         (raddr    ),
.WADDR         (waddr    ),
.WCLK          (wclk     ),
.WE            (we       )
);
```

6. Detailed function description

The GTP realizes the function of dual-port RAM, and the read and write addresses are independent of each other. Only when WE is high level, the input data is written into RAM. When WE is high level, the input data DI is written into the write address WADDR at the rising edge of WCLK In, the data in the read address RADDR is read from the DO port. When WE is low, the data in the read address RADDR is read from the DO port at the rising edge of WCLK, and no write operation is performed.

Reading and writing are performed asynchronously, and the read and write addresses cannot be the same at the same time to prevent read and write conflicts. As shown in the figure below.

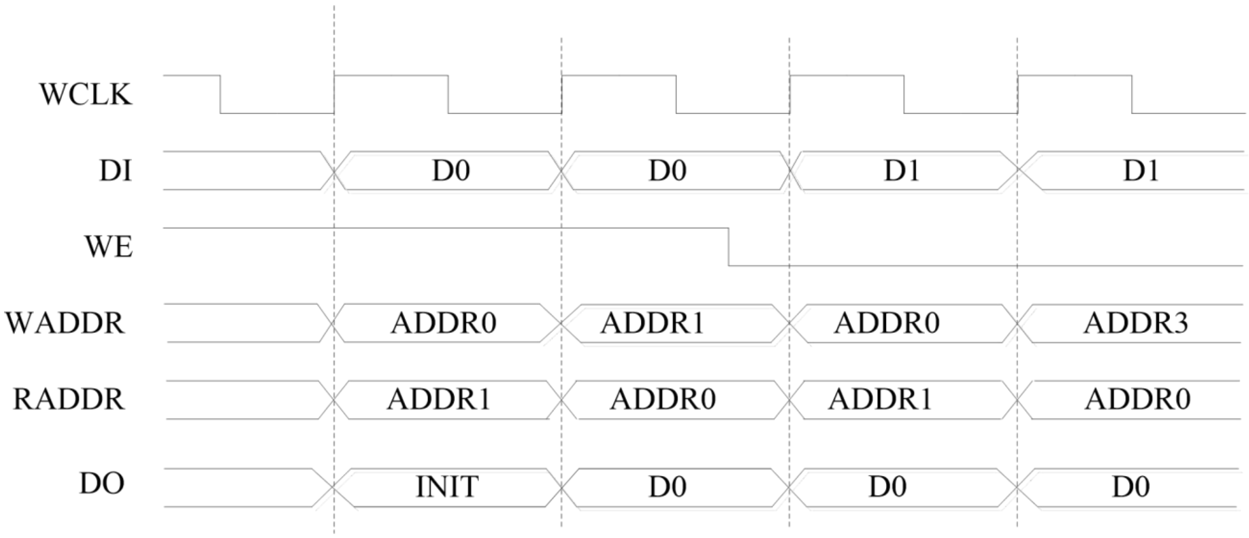


Figure 56 GTP_RAM128X1DP waveform diagram

(15) GTP_RAM256X1SP instruction

1. Supported devices

Table 140 GTP_RAM256X1SP device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_RAM256X1SP is a dual-port random memory module with a data bit width of 1 and an address depth of 256. The read and write addresses are input by two independent ports. The block diagram is as follows.

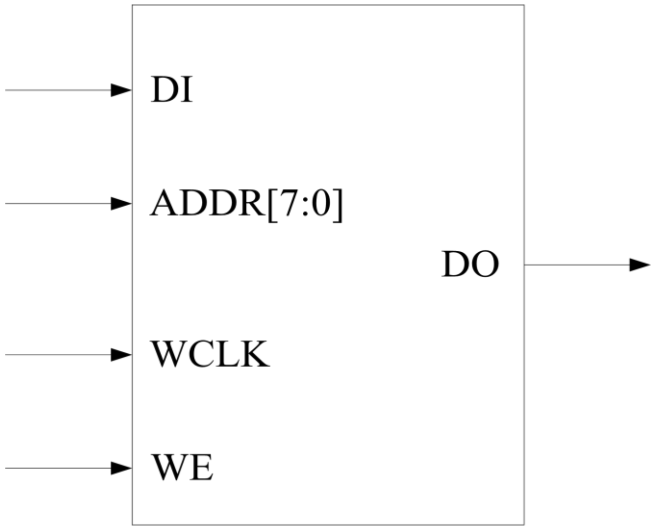


Figure 57 GTP_RAM256X1SP structure diagram

3. Port description

Table 141 GTP_RAM256X1SP port list

Port Name	Direction	Functional description
DI	Input	Write data
ADDR	Input	Write address
WCLK	Input	Write clock
WE	Input	Write enable
DO	Output	Read data

4. Parameter description

Table 142 GTP_RAM256X1SP parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
INIT	<binary>	256'h0~256'hffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff	128'h0	Memory initialization configuration parameter

5. Instantiation template

```
GTP_RAM256X1SP#(
.INIT          (256'h0)
) GTP_RAM256X1SP_inst(
.DO            (do       ),
.DI            (di       ),
.ADDR          (addr     ),
.WCLK          (wclk     ),
.WE            (we       )
);
```

6. Detailed function description

The GTP realizes the single-port RAM function, and only writes when WE is high. When WE is high, the input data DI is written into the current address ADDR at the rising edge of WCLK, and the input data is read from the DO port. Output. When WE is low level, the data in the current address ADDR is read from the DO port at the rising edge of WCLK, and no write operation is performed. As shown in the figure below.

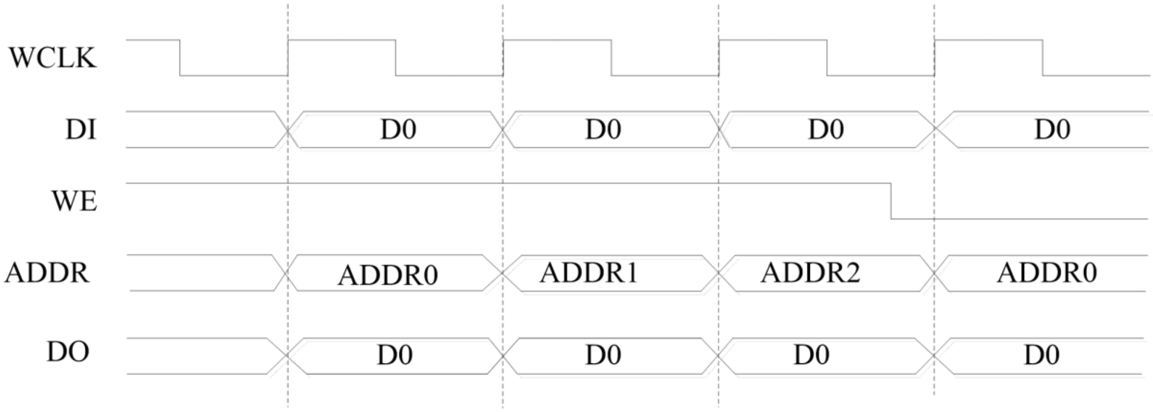


Figure 58 GTP_RAM256X1SP waveform diagram

(16) GTP_DRM36K_E1 instruction

1. Supported devices

Table 143 GTP_DRM36K_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DRM36K_E1 has 36K bits storage unit, which supports multiple working modes, including dual-port (DP, True Dual Port) RAM, simple dual-port (SDP, Simple Dual Port) RAM, single-port (SP, Single Port) RAM or ROM mode, DRM supports configurable data bit width, and supports dual-port mixed data bit width in DP RAM and SDP RAM modes. The structural block diagram is shown below.

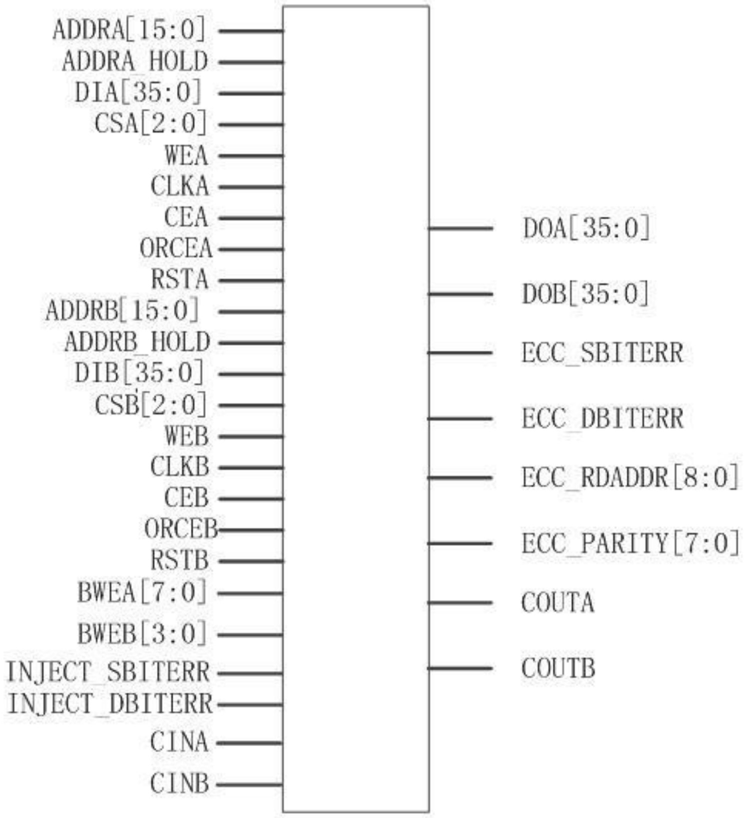


Figure 59 GTP_DRM36K_E1 structure diagram

3. Port description

Table 144 GTP_DRM36K_E1 port list

Port Name	Direction	Functional description
ADDRA	Input	A Port input address
ADDRA_HOLD	Input	A Port address input selection
DIA	Input	A Port data input
CSA	Input	A Port address extension
WEA	Input	A Port write enable
BWEA	Input	A Port Byte Write Enable
CLKA	Input	A Port clock
CEA	Input	A Port clock enable
ORCEA	Input	A Port output register enable
RSTA	Input	A Port data register reset
DOA	Output	A Port data output
ADDRB	Input	B Port input address
ADDRB_HOLD	Input	B Port address input selection
DIB	Input	B Port data input
CSB	Input	B Port address extension
WEB	Input	B Port write enable
BWEB	Input	B Port Byte Write Enable
CLKB	Input	B Port clock
CEB	Input	B Port clock enable
ORCEB	Input	B Port output register enable
RSTB	Input	BPort data register reset

Port Name	Direction	Functional description
CINA	Input	Adjacent DRM A port data output cascade input
CINB	Input	Adjacent DRM B port data output cascade input
INJECT_SBITERR	Input	ECC mode single bit error insertion
INJECT_DBITERR	Input	ECC mode double bit error insertion
DOB	Output	B port data output
COUTA	Output	Adjacent DRM A port data output cascade output
COUTB	Output	Adjacent DRM B port data output Cascade output
ECC_SBITERR	Output	ECC mode single bit error flag
ECC_DBITERR	Output	ECC mode double bits error flag
ECC_PARITY[7:0]	Output	ECC code check digit output
ECC_RDADDR[8:0]	Output	ECC decoding read address output

4. Parameter description

Table 145 GTP_DRM36K_E1 parameter list

Parameter Name	Functional description	Settings	Defaults
CSA_MASK[2:0]	A Port address extension control signal	0 ~ 7	3'b000
CSB_MASK[2:0]	B Port address expansion control signal	0 ~ 7	3'b000
DATA_WIDTH_A	A Maximum bit width of port data	1, 2, 4, 8, 16, 32, 9, 18, 36, 64, 72	18
DATA_WIDTH_B	B Maximum bit width of port data	1, 2, 4, 8, 16, 32, 9, 18, 36, 64, 72	18
WRITE_MODE_A	A Port write mode	NORMAL_WRITE TRANSPARENT_WRITE READ_BEFORE_WRITE	"NORMAL_WRITE"
WRITE_MODE_B	B Port write mode	NORMAL_WRITE TRANSPARENT_WRITE READ_BEFORE_WRITE	"NORMAL_WRITE"
DOA_REG	A Port output register	0: Disable output register 1: Enable output register	0
DOB_REG	B Port output register	0: Disable output register 1: Enable output register	0
RST_TYPE	Reset mode selection	SYNC: Synchronous reset ASYNC: Asynchronous reset	"SYNC"
RAM_MODE	RAM MODE SELECTION	TRUE_DUAL_PORT SIMPLE_DUAL_PORT RAM SINGLE_PORT ROM	"TRUE_DUAL_PORT"
GRS_EN	Global reset enable signal (chip internal)	FALSE: Do not enable global reset; TRUE: Enable global reset.	"TRUE"
DOA_REG_CLKINV	A Port output register clock toggle	0:clock not turning over 1:clock flip	0
DOB_REG_CLKINV	B Port output register clock toggle	0:clock not turning over 1:clock flip	0
RSTA_VAL	A Port output set/reset value	36'h0_0000_0000~36'hF_FFFF_FFFF	36'b0
RSTB_VAL	B Port output set/reset value	36'h0_0000_0000~36'hF_FFFF_FFFF	36'b0
RAM_CASCADE	64Kx1 hard cascade mode	"NONE": No hard cascading	"NONE"

Parameter Name	Functional description	Settings	Defaults
		"UPPER": As a hard cascade data output module "LOWER": As a hard cascade add-on module	
ECC_WRITE_EN	ECC write mode enable	"FALSE": Enable "TRUE": Disable	"FALSE"
ECC_READ_EN	ECC read mode enable	"FALSE": Enable "TRUE": Disable	"FALSE"
INIT_00 INIT_01 INIT_02 … INIT_7F	RAM initialization configuration parameters	0 ~2^288-1	288'h0
INIT_FILE	Initialization file path, this parameter is only valid in behavioral simulation, the initialization data of RAM comes from initialization when synthesizing netlist generation Configuration parameters INIT_XX	"NONE": If no initialization file is specified, the initialization data is the value set by the parameter INIT_XX; "XXX": XXX indicates the specific initialization file path, and the parameter INIT_XX is overwritten in the behavior simulation as Initial value of DRM	"NONE"
BLOCK_X	Data cascading coordinates during DRM36K cascading, used to map the same initialization file to different cascading DRMs	Depends on cascading DRM coordinates	0
BLOCK_Y	Address cascading coordinates during DRM36K cascading, used to map the same initialization file to different In cascading DRM	Depends on cascading DRM coordinates	0
RAM_DATA_WIDTH	The maximum bit width of data after DRM cascading is used to map the same initialization file to different levels Linked DRM	Depends on the number of cascaded DRMs	18
RAM_ADDR_WIDTH	The maximum bit width of the address after DRM cascading is used to map the same initialization file to different levels Linked DRM	Depends on the number of cascaded DRMs	11
INIT_FORMAT	Initialization file format	"BIN": Binary "HEX": hexadecimal	"BIN"

Note: SDP mode does not support read-write mode setting, when using GTP to configure DRM as SDP mode, you cannot manually modify and set the read-write mode parameter WRITE_MODE_A/B.Instantiation template

```
GTP_DRM36K_E1 #(
.GRS_EN("TRUE"),
.CSA_MASK(3'b0),
.CSB_MASK(3'b0),
.DATA_WIDTH_A(18),
.DATA_WIDTH_B(18),
.WRITE_MODE_A("NORMAL_WRITE"),
.WRITE_MODE_B("NORMAL_WRITE"),
```

```
.DOA_REG(0),
.DOB_REG(0),
.DOA_REG_CLKINV(0),
.DOB_REG_CLKINV(0),
.RSTA_VAL(36'b0),
.RSTB_VAL(36'b0),
.RST_TYPE("SYNC"),
.RAM_MODE("TRUE_DUAL_PORT"),
.RAM_CASCADE("NONE"),
.ECC_READ_EN("FALSE"),
.ECC_WRITE_EN("FALSE"),
.INIT_00(288'b0),
.INIT_01(288'b0),
.INIT_02(288'b0),
.INIT_03(288'b0),
.INIT_04(288'b0),
.INIT_05(288'b0),
.INIT_06(288'b0),
.INIT_07(288'b0),
.INIT_08(288'b0),
.INIT_09(288'b0),
.INIT_0A(288'b0),
.INIT_0B(288'b0),
.INIT_0C(288'b0),
.INIT_0D(288'b0),
.INIT_0E(288'b0),
.INIT_0F(288'b0),
.INIT_10(288'b0),
.INIT_11(288'b0),
.INIT_12(288'b0),
.INIT_13(288'b0),
.INIT_14(288'b0),
.INIT_15(288'b0),
.INIT_16(288'b0),
.INIT_17(288'b0),
.INIT_18(288'b0),
.INIT_19(288'b0),
.INIT_1A(288'b0),
.INIT_1B(288'b0),
.INIT_1C(288'b0),
.INIT_1D(288'b0),
.INIT_1E(288'b0),
.INIT_1F(288'b0),
.INIT_20(288'b0),
.INIT_21(288'b0),
.INIT_22(288'b0),
.INIT_23(288'b0),
.INIT_24(288'b0),
```

```
.INIT_25(288'b0),
.INIT_26(288'b0),
.INIT_27(288'b0),
.INIT_28(288'b0),
.INIT_29(288'b0),
.INIT_2A(288'b0),
.INIT_2B(288'b0),
.INIT_2C(288'b0),
.INIT_2D(288'b0),
.INIT_2E(288'b0),
.INIT_2F(288'b0),
.INIT_30(288'b0),
.INIT_31(288'b0),
.INIT_32(288'b0),
.INIT_33(288'b0),
.INIT_34(288'b0),
.INIT_35(288'b0),
.INIT_36(288'b0),
.INIT_37(288'b0),
.INIT_38(288'b0),
.INIT_39(288'b0),
.INIT_3A(288'b0),
.INIT_3B(288'b0),
.INIT_3C(288'b0),
.INIT_3D(288'b0),
.INIT_3E(288'b0),
.INIT_3F(288'b0),
.INIT_40(288'b0),
.INIT_41(288'b0),
.INIT_42(288'b0),
.INIT_43(288'b0),
.INIT_44(288'b0),
.INIT_45(288'b0),
.INIT_46(288'b0),
.INIT_47(288'b0),
.INIT_48(288'b0),
.INIT_49(288'b0),
.INIT_4A(288'b0),
.INIT_4B(288'b0),
.INIT_4C(288'b0),
.INIT_4D(288'b0),
.INIT_4E(288'b0),
.INIT_4F(288'b0),
.INIT_50(288'b0),
.INIT_51(288'b0),
.INIT_52(288'b0),
.INIT_53(288'b0),
.INIT_54(288'b0),
```

```
    .INIT_55(288'b0),
    .INIT_56(288'b0),
    .INIT_57(288'b0),
    .INIT_58(288'b0),
    .INIT_59(288'b0),
    .INIT_5A(288'b0),
    .INIT_5B(288'b0),
    .INIT_5C(288'b0),
    .INIT_5D(288'b0),
    .INIT_5E(288'b0),
    .INIT_5F(288'b0),
    .INIT_60(288'b0),
    .INIT_61(288'b0),
    .INIT_62(288'b0),
    .INIT_63(288'b0),
    .INIT_64(288'b0),
    .INIT_65(288'b0),
    .INIT_66(288'b0),
    .INIT_67(288'b0),
    .INIT_68(288'b0),
    .INIT_69(288'b0),
    .INIT_6A(288'b0),
    .INIT_6B(288'b0),
    .INIT_6C(288'b0),
    .INIT_6D(288'b0),
    .INIT_6E(288'b0),
    .INIT_6F(288'b0),
    .INIT_70(288'b0),
    .INIT_71(288'b0),
    .INIT_72(288'b0),
    .INIT_73(288'b0),
    .INIT_74(288'b0),
    .INIT_75(288'b0),
    .INIT_76(288'b0),
    .INIT_77(288'b0),
    .INIT_78(288'b0),
    .INIT_79(288'b0),
    .INIT_7A(288'b0),
    .INIT_7B(288'b0),
    .INIT_7C(288'b0),
    .INIT_7D(288'b0),
    .INIT_7E(288'b0),
    .INIT_7F(288'b0),
    .INIT_FILE("NONE"),
    .BLOCK_X(0),
    .BLOCK_Y(0),
    .RAM_DATA_WIDTH(9),
    .RAM_ADDR_WIDTH(12),
```

```
.INIT_FORMAT("BIN")
) GTP_DRM36K_E1_inst (

.DOA                    (doa      ),
.DOB                    (dob      ),
.ECC_PARITY             (ecc_parity),
.ECC_RDADDR             (ecc_rdaddr),
.ADDRA                  (addra    ),
.ADDRB                  (addrb    ),
.BWEA                   (bwea     ),
.BWEB                   (bweb     ),
.CSA                    (csa      ),
.CSB                    (csb      ),
.DIA                    (dia      ),
.DIB                    (dib      ),
.COUTA                  (couta    ),
.COUTB                  (coutb    ),
.ECC_DBITERR            (ecc_dbiterr),
.ECC_SBITERR            (ecc_sbiterr),
.ADDRA_HOLD             (addra_hold),
.ADDRB_HOLD             (addrb_hold),
.CEA                    (cea      ),
.CEB                    (ceb      ),
.CINA                   (cina     ),
.CINB                   (cinb     ),
.CLKA                   (clka     ),
.CLKB                   (clkb     ),
.INJECT_DBITERR         (inject_dbiterr),
.INJECT_SBITERR         (inject_sbiterr),
.ORCEA                  (orcea    ),
.ORCEB                  (orceb    ),
.RSTA                   (rsta     ),
.RSTB                   (rstb     ),
.WEA                    (wea      ),
.WEB                    (web      )
);
```

(17) GTP_DRM18K_E1 instruction

1. Supported devices

Table 146 GTP_DRM18K_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_DRM18K_E1 has 18K bits storage unit, which supports multiple working modes, including dual-port (DP, True Dual Port) RAM, simple dual-port (SDP, Simple Dual Port) RAM, single-port (SP, Single Port) RAM or ROM mode,

DRM supports configurable data bit width, and supports dual-port mixed data bit width in DP RAM and SDP RAM modes. The block diagram is as follows:

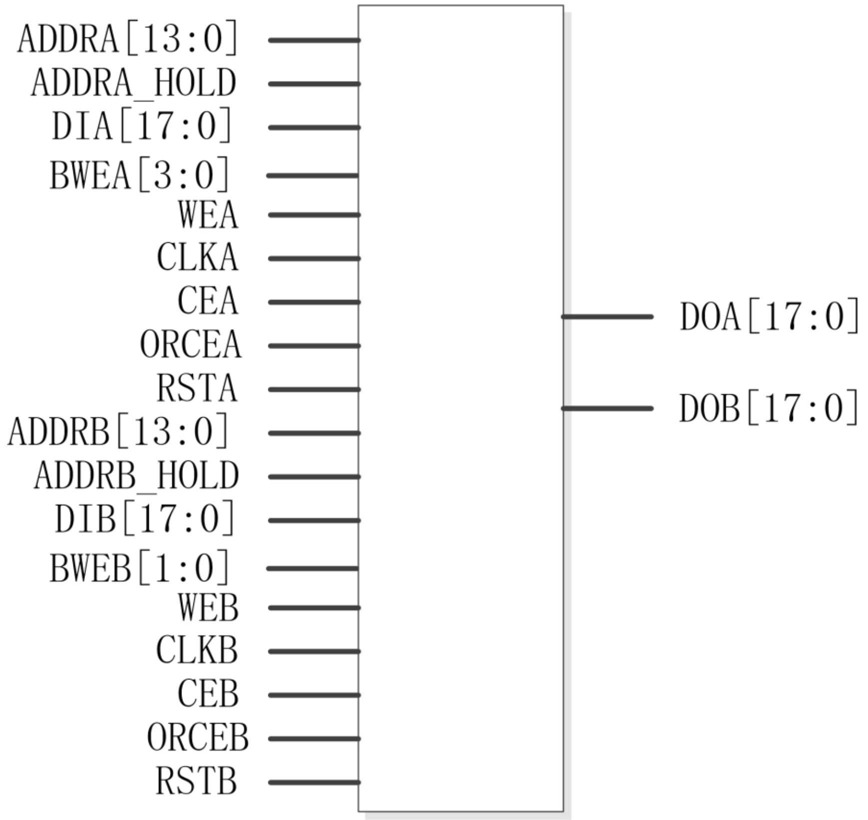


Figure 60 GTP_DRM18K_E1 structure diagram

3. Port description

Table 147 GTP_DRM18K_E1 port list

Port Name	Direction	Functional description
ADDRA	Input	A Port input address
ADDRA_HOLD	Input	A Port address input selection
DIA	Input	A Port data input
WEA	Input	A Port write enable
CLKA	Input	A Port clock
CEA	Input	A Port clock enable
ORCEA	Input	A Port output register enable
RSTA	Input	A Port data register reset
BWEA	Input	A Port Byte Write Enable
DOA	Output	A Port data output
ADDRB	Input	B Port input address
ADDRB_HOLD	Input	B Port address input selection
DIB	Input	B Port data input
WEB	Input	B Port write enable
CLKB	Input	B Port clock
CEB	Input	B Port clock enable
ORCEB	Input	B Port output register enable
RSTB	Input	BPort data register reset

Port Name	Direction	Functional description
BWEB	Input	B Port Byte Write Enable
DOB	Output	B Port data output

4. Parameter description

Table 148 GTP_DRM18K_E1 parameter list

Parameter Name	Functional description	Settings	Defaults
DATA_WIDTH_A	A Maximum bit width of port data	1, 2, 4, 8, 16, 32, 9, 18, 36	18
DATA_WIDTH_B	B Maximum bit width of port data	1, 2, 4, 8, 16, 32, 9, 18, 36	18
WRITE_MODE_A	A Port write mode	NORMAL_WRITE TRANSPARENT_WRITE READ_BEFORE_WRITE	NORMAL_WRITE
WRITE_MODE_B	B Port write mode	NORMAL_WRITE TRANSPARENT_WRITE READ_BEFORE_WRITE	NORMAL_WRITE
DOA_REG	A Port output register	0: Disable output register 1: Enable output register	0
DOB_REG	B Port output register	0: Disable output register 1: Enable output register	0
RST_TYPE	Reset mode selection	SYNC: Synchronous reset ASYNC: Asynchronous reset	SYNC
RSTA_VAL	A Port output set/reset value	18'h0_0000~18'h3_FFFF	18'b0
RSTB_VAL	B Port output set/reset value	18'h0_0000~18'h3_FFFF	18'b0
RAM_MODE	RAM mode selection	TRUE_DUAL_PORT: dual-port RAM SIMPLE_DUAL_PORT: simple dual-port RAM SINGLE_PORT: single port RAM ROM: ROM	TRUE_DUAL_PORT
GRS_EN	Global reset enable signal (on-chip department)	FALSE: Do not enable global reset; TRUE: Enable global reset.	TRUE
DOA_REG_CLKINV	PORT OUTPUT REGISTER CLOCK TOGGLE	0:clock not turning over 1:clock flip	0
DOB_REG_CLKINV	B Port output register clock toggle	0:clock not turning over 1:clock flip	0
INIT_00 INIT_01 INIT_02 … INIT_3F	RAM initialization configuration parameters	0~2^288-1	0
INIT_FILE	Initialization file path, this parameter is only valid in behavioral simulation, the initialization data of RAM comes from the initialization configuration parameters when synthesizing the netlist generation INIT_XX	"NONE": If no initialization file is specified, the initialization data is the value set by the parameter INIT_XX; "XXX": XXX indicates the specific initialization file path, and the parameter INIT_XX is overridden in the behavior simulationis the initial value of DRM	NONE
BLOCK_X	Data cascading coordinates during DRM36K cascading, used to map the same initialization file to different cascading DRMs	Depends on cascading DRM coordinates	0

Parameter Name	Functional description	Settings	Defaults
BLOCK_Y	DRM36K address cascade coordinates when cascading, used to map the same initialization file to different cascaded DRM	Depends on the coordinates of the cascade	0
RAM_DATA_WIDTH	The maximum bit width of the data after the DRM cascade is used to map the same initialization file to different cascaded DRM	Depends on the number of cascaded DRM	9
RAM_ADDR_WIDTH	The maximum bit width of the address after the DRM cascade, which is used to map the same initialization file to different cascaded DRM	Depends on the number of cascaded DRM	11
INIT_FORMAT	Initialization file format	"BIN": binary "HEX": Hexadecimal	BIN

Note: SDP mode does not support read-write mode settings. When using GTP to configure DRM to SDP mode, you cannot manually modify the read-write mode parameter WRITE_MODE_A/B.

5. Instantiation template

```
GTP_DRM18K_E1 #(
.GRS_EN("TRUE"),
.DATA_WIDTH_A(18),
.DATA_WIDTH_B(18),
.DOA_REG(0),
.DOB_REG(0),
.DOA_REG_CLKINV(0),
.DOB_REG_CLKINV(0),
.RSTA_VAL(18'b0),
.RSTB_VAL(18'b0),
.RST_TYPE("SYNC"),
.RAM_MODE("TRUE_DUAL_PORT"),
.WRITE_MODE_A("NORMAL_WRITE"),
.WRITE_MODE_B("NORMAL_WRITE"),
.INIT_00(288'b0),
.INIT_01(288'b0),
.INIT_02(288'b0),
.INIT_03(288'b0),
.INIT_04(288'b0),
.INIT_05(288'b0),
.INIT_06(288'b0),
.INIT_07(288'b0),
.INIT_08(288'b0),
.INIT_09(288'b0),
.INIT_0A(288'b0),
.INIT_0B(288'b0),
.INIT_0C(288'b0),
.INIT_0D(288'b0),
```

```
.INIT_0E(288'b0),
.INIT_0F(288'b0),
.INIT_10(288'b0),
.INIT_11(288'b0),
.INIT_12(288'b0),
.INIT_13(288'b0),
.INIT_14(288'b0),
.INIT_15(288'b0),
.INIT_16(288'b0),
.INIT_17(288'b0),
.INIT_18(288'b0),
.INIT_19(288'b0),
.INIT_1A(288'b0),
.INIT_1B(288'b0),
.INIT_1C(288'b0),
.INIT_1D(288'b0),
.INIT_1E(288'b0),
.INIT_1F(288'b0),
.INIT_20(288'b0),
.INIT_21(288'b0),
.INIT_22(288'b0),
.INIT_23(288'b0),
.INIT_24(288'b0),
.INIT_25(288'b0),
.INIT_26(288'b0),
.INIT_27(288'b0),
.INIT_28(288'b0),
.INIT_29(288'b0),
.INIT_2A(288'b0),
.INIT_2B(288'b0),
.INIT_2C(288'b0),
.INIT_2D(288'b0),
.INIT_2E(288'b0),
.INIT_2F(288'b0),
.INIT_30(288'b0),
.INIT_31(288'b0),
.INIT_32(288'b0),
.INIT_33(288'b0),
.INIT_34(288'b0),
.INIT_35(288'b0),
.INIT_36(288'b0),
.INIT_37(288'b0),
.INIT_38(288'b0),
.INIT_39(288'b0),
.INIT_3A(288'b0),
.INIT_3B(288'b0),
.INIT_3C(288'b0),
.INIT_3D(288'b0),
```

```
.INIT_3E(288'b0),
.INIT_3F(288'b0),
.INIT_FILE("NONE"),
.BLOCK_X(0),
.BLOCK_Y(0),
.RAM_DATA_WIDTH(9),
.RAM_ADDR_WIDTH(11),
.INIT_FORMAT("BIN")
) GTP_DRM18K_E1_inst (
.DOA          (doa      ),
.DOB          (dob      ),
.ADDRA        (addra    ),
.ADDRB        (addrb    ),
.BWEA         (bwea     ),
.BWEB         (bweb     ),
.DIA          (dia      ),
.DIB          (dib      ),
.ADDRA_HOLD   (addra_hold),
.ADDRB_HOLD   (addrb_hold),
.CEA          (cea      ),
.CEB          (ceb      ),
.CLKA         (clka     ),
.CLKB         (clkb     ),
.ORCEA        (orcea    ),
.ORCEB        (orceb    ),
.RSTA         (rsta     ),
.RSTB         (rstb     ),
.WEA          (wea      ),
.WEB          (web      )
);
```

(18) GTP_FIFO36K_E1 instruction

1. Supported devices

Table 149 GTP_FIFO36K_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_FIFO36K_E1 supports 36K bits storage unit, which can be configured as 32K*1, 16K*2, 8K*4, 4K*9(8), 2K*18(16), 1K*36(32), 512*72(64) Data bit width mode; GTP_FIFO36K_E1 does not support mixed bit width and Byte enable write operations.

When using GTP_FIFO36K_E1, DRM is configured in SDP mode, and does not support mixed bit widths, and does not support Byte enable write operations. One port of DRM is dedicated to FIFO data writing, and the other port is dedicated to FIFO data reading. In asynchronous FIFO mode, the read and write ports can use different clocks.

Its structural block diagram is as follows:

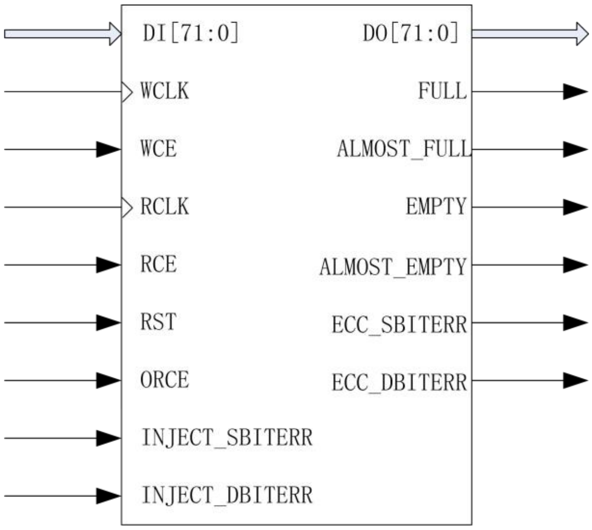


Figure 61 GTP_FIFO36K structure diagram

3. Port description

Table 150 GTP_FIFO36K_E1 port list

Port Name	Direction	Functional description
DI	Input	Data writing
WCLK	Input	Write clock signal
RCLK	Input	Read clock signal
WCE	Input	Write CE signal
RCE	Input	Read CE signal
RST	Input	Reset
ORCE	Input	Output register enable signal
INJECT_SBITERR	Input	ECC mode single bit error insertion
INJECT_DBITERR	Input	ECC mode double-bit error insertion
DO	Output	Data read out
EMPTY	Output	Read port read empty flag
FULL	Output	Write port full flag
ALMOST_EMPTY	Output	The read port will be empty.
ALMOST_FULL	Output	Write port will be full flag
ECC_SBITERR	Output	ECC mode single-bit error flag
ECC_DBITERR	Output	ECC mode dual bits error flag

4. Parameter description

Table 151 GTP_FIFO36K_E1 parameter list

Parameter Name	Functional description	Settings	Defaults
GRS_EN	Global reset enable signal (chip internal)	TRUE: Enable global reset; FALSE: Do not enable global reset.	TRUE
DATA_WIDTH	FIFO Data bit width	1, 2, 4, 8, 9, 16, 18, 32, 36, 64, 72	18
ALMOST_FULL_OFFSET	When the FIFO will be full and the difference between the position of the write pointer and the read pointer is equal to	DATA_WIDTH=1bit:	15'h0000

Parameter name	Function description	Set value	Default value
	ALMOST_FULL_OFFSET, generates the almost_full flag	sync FIFO :1~32767 async FIFO :1~32764 2bits: sync FIFO :1~16383 async FIFO:1~16380 4bits: sync FIFO :1~8191 async FIFO:1~8188 8/9bits: sync FIFO :1~4095 async FIFO: 1~4092 16/18bits: sync FIFO :1~2047 async FIFO :1~2044 32/36bits: sync FIFO : 1~1023 async FIFO : 1~1020 64/72bits: sync FIFO : 1~511 async FIFO : 1~508	
ALMOST_EMPTY_OFFSET	When the FIFO is empty and the difference between the positions of the read pointer and the write pointer is equal to ALMOST_EMPTY_OFFSET, the almost_empty flag is generated.	DATA_WIDTH= 1bit: sync FIFO :1~32767 async FIFO :4~32767 2bits: sync FIFO :1~16383 async FIFO: 4~16383 4bits: sync FIFO :1~8191 async FIFO: 4~8191 8/9bits: sync FIFO :1~4095 async FIFO: 4~4095 16/18bits: sync FIFO :1~2047 async FIFO : 4~2047 32/36bits: sync FIFO : 1~1023 async FIFO : 4~1023 64/72bits: sync FIFO : 1~511 async FIFO : 4~511	15’h0000
SYNC_FIFO	Asynchronous/synchronous FIFO selection	TRUE: USE synchronous FIFO； FALSE: Use asynchronous FIFO.	FALSE
USE_EMPTY	Enable empty read flag	1: Enable empty read flag; 0: Do not enable the blank reading flag.	0
USE_FULL	Enable full flag	1: Enable full flag; 0: Do not enable the full flag.	0

Parameter Name	Functional description	Settings	Defaults
DO_REG	Enable output register enable	1: Enable; 0: Disable	0
ECC_WRITE_EN	Write port ECC mode is enabled	TRUE: Enable; FALSE: Disable.	FALSE
ECC_READ_EN	Read port ECC mode is enabled	TRUE: Enable; FALSE: Disable.	FALSE
RST_VAL	Output set/reset value	72'h00_0000_0000_0000_0000~ 72'hFF_FFFF_FFFF_FFFF_FFFF	72'b0

5. Instantiation template

```
GTP_FIFO36K_E1 #(
.GRS_EN("TRUE"),
.DATA_WIDTH(18),
.DO_REG(0),
.ECC_READ_EN("FALSE"),
.ECC_WRITE_EN("FALSE"),
.ALMOST_FULL_OFFSET(15'b0),
.ALMOST_EMPTY_OFFSET(15'b0),
.RST_VAL(72'b0),
.USE_EMPTY(0),
.USE_FULL(0),
.SYNC_FIFO("FALSE")
) GTP_FIFO36K_E1_inst (
.DO                     (do                 ),
.DI                     (di                 ),
.ALMOST_EMPTY           (almost_empty       ),
.ALMOST_FULL            (almost_full        ),
.ECC_DBITERR            (ecc_dbiterr        ),
.ECC_SBITERR            (ecc_sbiterr        ),
.EMPTY                  (empty              ),
.FULL                   (full               ),
.INJECT_DBITERR         (inject_dbiterr     ),
.INJECT_SBITERR         (inject_sbiterr     ),
.ORCE                   (orce               ),
.RCE                    (rce                ),
.RCLK                   (rclk               ),
.RST                    (rst                ),
.WCE                    (wce                ),
.WCLK                   (wclk               )
);
```

(19) GTP_FIFO18K_E1 instruction

1. Supported devices

Table 152 GTP_FIFO18K_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_FIFO18K_E1 supports 18K bits storage unit, which can be configured as 16K*1, 8K*2, 4K*4, 2K*9(8), 1K*18(16) data bit width mode;

When using GTP_FIFO18K_E1, DRM is configured in SDP mode, and does not support mixed bit widths, and does not support Byte enable write operations. One port of DRM is dedicated to FIFO data writing, and the other port is dedicated to FIFO data reading. In asynchronous FIFO mode, the read and write ports can use different clocks;

The block diagram is as follows:

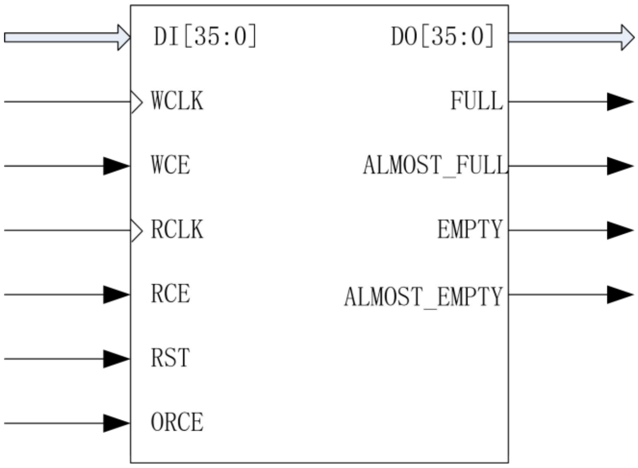


Figure 62 GTP_FIFO18K structure diagram

3. Port description

Table 153 GTP_FIFO18K_E1 port list

Port Name	Direction	Functional description
ALMOST_EMPTY	Output	The read port will be empty.
ALMOST_FULL	Output	Write port will be full flag
EMPTY	Output	Read port read empty flag
FULL	Output	Write port full flag
DO[35:0]	Output	Data read out
DI[35:0]	Input	Data writing
WCLK	Input	Write clock
RCLK	Input	Read the clock
WCE	Input	Write enable
RCE	Input	Read enable
ORCE	Input	Output register enable signal

Port Name	Direction	Functional description
RST	Input	Reset, high level is valid

4. Parameter description

Table 154 GTP_FIFO18K_E1 parameter list

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	string	"TRUE" "FALSE"	"TRUE"	Global reset is enabled
DATA_WIDTH	integer	1, 2,4,8,9,16,18	18	FIFO DATA BIT WIDTH
DO_REG	integer	0, 1	0	The output register is enabled 1: Enable 0: Disable
ALMOST_FULL_OFFSET	binary	14'h0000 – 14'hFFFF	14'h0	DATA_WIDTH= 1bit: sync FIFO :1~16383 async FIFO:1~16380 2bits: sync FIFO :1~8191 async FIFO:1~8188 4bits: sync FIFO :1~4095 async FIFO: 1~4092 8/9bits: sync FIFO :1~2047 async FIFO :1~2044 16/18bits: sync FIFO : 1~1023 async FIFO : 1~1020 32/36bits: sync FIFO : 1~511 async FIFO : 1~508
ALMOST_EMPTY_OFFSE T	binary	14'h0000 – 14'hFFFF	14'h0	DATA_WIDTH= 1bit: sync FIFO :1~16383 async FIFO: 4~16383 2bits: sync FIFO :1~8191 async FIFO: 4~8191 4bits: sync FIFO :1~4095 async FIFO: 4~4095 8/9bits: sync FIFO :1~2047 async FIFO : 4~2047 16/18bits: sync FIFO : 1~1023 async FIFO : 4~1023 32/36bits: sync FIFO : 1~511 async FIFO : 4~511

Parameter Name	Parameter Type	Settings	Defaults	Functional description
RST_VAL	binary	36'h0 – 36'hF_FFFF_FFFF	36'h0	Output reset/set value
USE_EMPTY	integer	0, 1	0	1: Enable blank reading flag 0: Do not enable the blank reading flag
USE_FULL	integer	0, 1	0	1: Enable full flag 0: Do not enable the full flag
SYNC_FIFO	string	"TRUE" "FALSE"	"FALSE"	"TRUE": Enable synchronous FIFO "FALSE": Enable asynchronous FIFO

5. Instantiation template

```
GTP_FIFO18K_E1 #(
.GRS_EN("TRUE"),
.DATA_WIDTH(18),
.DO_REG(0),
.ALMOST_FULL_OFFSET(15'b0),
.ALMOST_EMPTY_OFFSET(15'b0),
.RST_VAL(36'b0),
.USE_EMPTY(0),
.USE_FULL(0),
.SYNC_FIFO("FALSE")
) GTP_FIFO18K_E1_inst (
.DO                    (do         ),
.DI                    (di         ),
.ALMOST_EMPTY          (almost_empty),
.ALMOST_FULL           (almost_full ),
.EMPTY                 (empty      ),
.FULL                  (full       ),
.ORCE                  (orce       ),
.RCE                   (rce        ),
.RCLK                  (rclk       ),
.RST                   (rst        ),
.WCE                   (wce        ),
.WCLK                  (wclk       )
);
```

5. IO related GTP instructions

(1) GTP_INBUF instruction

1. Supported devices

Table 155 GTP_INBUF device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_INBUF is an input buffer. As shown below:

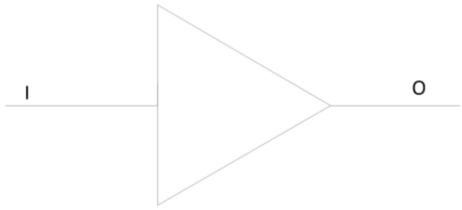


Figure 63 GTP_INBUF

3. Port description

Table 156 GTP_INBUF port list description

Port Name	Direction	Functional description
I	Input	Input signal
O	Output	Output signal

4. Parameter description

Table 157 GTP_INBUF parameter list

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See Table 158	DEFAULT=LVCMOS33	IO standard
TERM_DDR	See Table 158	"ON"	HSTL, SSTL standard input, built-in terminal resistor on or off

Table 158 GTP_INBUF list of valid values for the parameter

GTP_INBUF	
IOSTANDARD	TERM_DDR
LVTTL33	None
PCI33	

GTP_INBUF	
IOSTANDARD	TERM_DDR
LVCMOS33	
LVCMOS25	
LVCMOS18	
LVCMOS15	
LVCMOS12	
SSTL25_I	OFF/ON
SSTL25_II	OFF/ON
SSTL18_I	OFF/ON
SSTL18_II	OFF/ON
SSTL15_I	OFF/ON
SSTL15_II	OFF/ON
HSTL18_I	OFF/ON
HSTL18_II	OFF/ON
HSTL15_I	OFF/ON
SSTL15_I_CAL	ON
SSTL15_II_CAL	ON
HSTL15_I_CAL	ON
SSTL135_I	OFF/ON
SSTL135_II	OFF/ON
SSTL12	OFF/ON
HSTL12_I	OFF/ON
HSUL12	OFF/ON
POD12	OFF/ON
HSTL15_II	OFF/ON
SSTL18_I_CAL	ON
SSTL18_II_CAL	ON
SSTL15_I_CAL	ON
SSTL15_II_CAL	ON
HSTL18_I_CAL	ON
HSTL18_II_CAL	ON
HSTL15_I_CAL	ON
HSTL15_II_CAL	ON
SSTL135_I_CAL	ON
SSTL135_II_CAL	ON
SSTL12_CAL	ON
HSTL12_I_CAL	ON
HSUL12_CAL	ON
POD12_CAL	ON
LVCAL_15	None
LVCAL_18	
HSLVCAL_15	
HSLVCAL_18	

Note: It contains all the standards of HR and HP (that is, the list of valid values of common parameters of Titan2 and other series).

5. Instantiation template

```
GTP_INBUF#(
.IOSTANDARD ("DEFAULT"),
.TERM_DDR    ("ON")
)
GTP_INBUF_inst (
.I  (i),
.O  (o)
);
```

(2) GTP_INBUFG instruction

1. Supported devices

Table 159 GTP_INBUFG device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_INBUFG is a clock input buffer. As shown below:

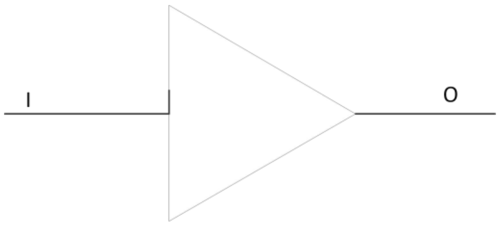


Figure 64 GTP_INBUFG

3. Port description

Table 160 GTP_INBUFG port list description

Port Name	Direction	Functional description
I	Input	Input signal
O	Output	Output signal

4. Parameter description

Table 161 GTP_INBUFG parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 162	DEFAULT=LVCMOS33	IO STANDARD
TERM_DDR	See table 162	"ON"	HSTL, SSTL standard input, built-in terminal resistor on or off

Table 162 GTP_INBUFG list of valid values for the parameter

GTP_INBUFG	
IOSTANDARD	TERM_DDR
LVCMOS33	NONE
LVCMOS25	
LVCMOS18	
LVCMOS15	
LVCMOS12	
SSTL25_I	OFF/ON
SSTL25_II	OFF/ON
SSTL18_I	OFF/ON
SSTL18_II	OFF/ON
SSTL15_I	OFF/ON
SSTL15_II	OFF/ON
HSTL18_I	OFF/ON
HSTL18_II	OFF/ON
HSTL15_I	OFF/ON
SSTL15_I_CAL	ON
SSTL15_II_CAL	ON
HSTL15_I_CAL	ON
SSTL135_I	OFF/ON
SSTL135_II	OFF/ON
SSTL12	OFF/ON
HSTL12_I	OFF/ON
HSUL12	OFF/ON
POD12	OFF/ON
HSTL15_II	OFF/ON
SSTL18_I_CAL	ON
SSTL18_II_CAL	ON
HSTL18_I_CAL	ON
HSTL18_II_CAL	ON
HSTL15_II_CAL	ON
SSTL135_I_CAL	ON
SSTL135_II_CAL	ON
SSTL12_CAL	ON
HSTL12_I_CAL	ON
HSUL12_CAL	ON
POD12_CAL	ON
LVCAL_15	NONE
LVCAL_18	
HSLVCAL_15	
HSLVCAL_18	

Note: It contains all the standards of HR and HP (that is, the list of valid values of common parameters of Titan2 and other series).

5. Instantiation template

```
GTP_INBUFG#(
.IOSTANDARD ("DEFAULT"),
. TERM_DDR ("ON")
)
GTP_INBUFG_inst (
.I   (i),
.O   (o)
);
```

(3) GTP_INBUFDS instruction

1. Supported devices

Table 163 GTP_INBUFDS device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_INBUFDS is a differential input buffer. As shown below:

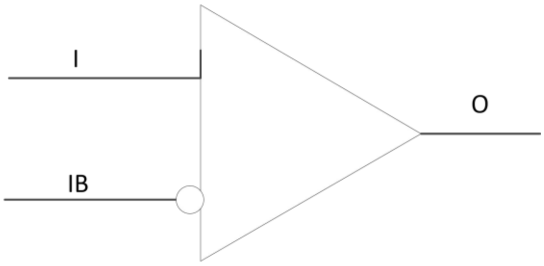


Figure 65 GTP_INBUFDS

3. Port description

Table 164 GTP_INBUFDS port list description

Port Name	Direction	Functional description
I	Input	Differential P input signal
IB	Input	Differential N input signal
O	Output	Output signal

4. Parameter description

Table 165 GTP_INBUFDS parameter list

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 166	DEFAULT=LVDS	IO STANDARD
TERM_DIFF	See table 166	"ON"	When differential input, the built-in terminal resistor is turned on or off

Table 166 GTP_INBUFDS list of valid values for the parameter (Titan2)

GTP_INBUFDS	
IOSTANDARD	TERM_DIFF
LVDS_18	OFF/ON
LVDS	OFF/ON
MINI-LVDS	OFF/ON
SUB-LVDS	OFF/ON
TMDS	OFF/ON
PPDS	OFF/ON
RSDS	OFF/ON

5. Instantiation template

```
GTP_INBUFDS#(
.IOSTANDARD ("DEFAULT"),
.TERM_DIFF("ON")
)
GTP_INBUFDS_inst (
.I  (i),
.IB (ib),
.O  (o)
);
```

(4) GTP_INBUFGDS instruction

1. Supported devices

Table 167 GTP_INBUFGDS device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_INBUFGDS is a differential clock input buffer. As shown below:

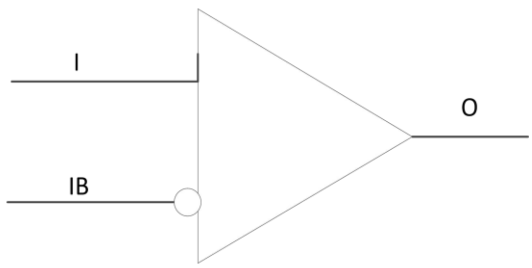


Figure 66 GTP_INBUFGDS

3. Port description

Table 168 GTP_INBUFGDS port list description

Port Name	Direction	Functional description
I	Input	Differential P input signal
IB	Input	Differential N input signal
O	Output	Output signal

4. Parameter description

Table 169 GTP_INBUFGDS parameter description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 170	DEFAULT=LVDS	IO STANDARD
TERM_DIFF	See table 170	"ON"	When differential input, the built-in terminal resistor is turned on or off

Table 170 GTP_INBUFGDS list of valid values for the parameter (Titan2)

GTP_INBUFGDS	
IOSTANDARD	TERM_DIFF
LVDS_18	OFF/ON
LVDS	OFF/ON
MINI-LVDS	OFF/ON
SUB-LVDS	OFF/ON
TMDS	OFF/ON
PPDS	OFF/ON
RSDS	OFF/ON

5. Instantiation template

```
GTP_INBUFGDS#(
.IOSTANDARD ("DEFAULT"),
.TERM_DIFF ("ON")
)
GTP_INBUFGDS_inst (
.I   (i),
.IB  (ib ),
.O   (o)
);
```

(5) GTP_INBUFE instruction

1. Supported devices

Table 171 GTP_INBUFE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_INBUFE supports single-ended input drive function.

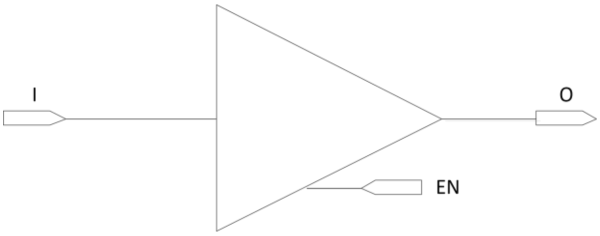


Figure 67 GTP_INBUFE

3. Port description

Table 172 GTP_INBUFE port list description

Port Name	Direction	Functional description
I	Input	Single-ended signal input
O	Output	Input buffer output, fed into the chip internal
EN	Input	Turn off the input buffer when it is 0.

4. Parameter description

Table 173 GTP_INBUFE parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 174	DEFAULT =LVCMOS33	Input IO standard
TERM_DDR	See table 174	"ON"	HSTL, SSTL standard input, built-in terminal resistor on or off

Table 174 GTP_INBUFE list of valid values for the parameter

GTP_INBUFE	
IOSTANDARD	TERM_DDR
LVTTL33	NONE
PCI33	
LVCMOS33	
LVCMOS25	
LVCMOS18	
LVCMOS15	
LVCMOS12	
SSTL25_I	OFF/ON
SSTL25_II	OFF/ON
SSTL18_I	OFF/ON
SSTL18_II	OFF/ON
SSTL15_I	OFF/ON

GTP_INBUFE	
IOSTANDARD	TERM_DDR
SSTL15_II	OFF/ON
HSTL18_I	OFF/ON
HSTL18_II	OFF/ON
HSTL15_I	OFF/ON
SSTL135_I	OFF/ON
SSTL135_II	OFF/ON
SSTL12	OFF/ON
HSTL12_I	OFF/ON
HSUL12	OFF/ON
POD12	OFF/ON
HSTL15_II	OFF/ON
SSTL18_I_CAL	ON
SSTL18_II_CAL	ON
SSTL15_I_CAL	ON
SSTL15_II_CAL	ON
HSTL18_I_CAL	ON
HSTL18_II_CAL	ON
HSTL15_I_CAL	ON
HSTL15_II_CAL	ON
SSTL135_I_CAL	ON
SSTL135_II_CAL	ON
SSTL12_CAL	ON
HSTL12_I_CAL	ON
HSUL12_CAL	ON
POD12_CAL	ON
LVCAL_15	NONE
LVCAL_18	
HSLVCAL_15	
HSLVCAL_18	

Note: Contains all the standards of HR and HP (that is, the list of effective values of common parameters of Titan2 and other series).

5. Instantiation template

```
GTP_INBUFE #(
.IOSTANDARD("DEFAULT"),
.TERM_DDR("ON")
) GTP_INBUFE_inst (
.O(),
.EN(),
.I()
);
```

(6) GTP_INBUFEDS instruction

1. Supported devices

Table 175 GTP_INBUFEDS device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_INBUFEDS supports input function. In addition to increasing the STANDBY signal, the input buffer can be turned off, and the others are consistent with GTP_INBUFDS.

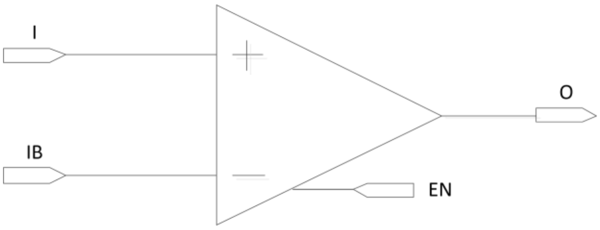


Figure 68 GTP_INBUFEDS

3. Port description

Table 176 GTP_INBUFEDS port list description

Port Name	Direction	Functional description
I	Input	Non-inverting differential input
IB	Input	Inverting differential input
O	Output	Differential output, sent to the chip
EN	Input	Turn off the input buffer when it is 0.

4. Parameter description

Table 177 GTP_INBUFEDS parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 178	DEFAULT =LVDS	Input IO standard
TERM_DIFF	See table 178	"ON"	For differential input, the built-in terminal resistor is turned on or off

Table 178 GTP_INBUFEDS list of valid values for the parameter (Titan2)

GTP_INBUFEDS	
IOSTANDARD	TERM_DIFF
LVDS_18	OFF/ON

GTP_INBUFEDS	
IOSTANDARD	TERM_DIFF
LVDS	OFF/ON
MINI-LVDS	OFF/ON
SUB-LVDS	OFF/ON
TMDS	OFF/ON
PPDS	OFF/ON
RSDS	OFF/ON

5. Instantiation template

```
GTP_INBUFEDS #(
.IOSTANDARD("DEFAULT"),
.TERM_DIFF("ON")
) GTP_INBUFEDS_inst (
.O(),
.EN(),
.I(),
.IB()
);
```

(7) GTP_OUTBUF instruction

1. Supported devices

Table 179 GTP_OUTBUF device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_OUTBUF is an output BUFFER. As shown in the figure below:

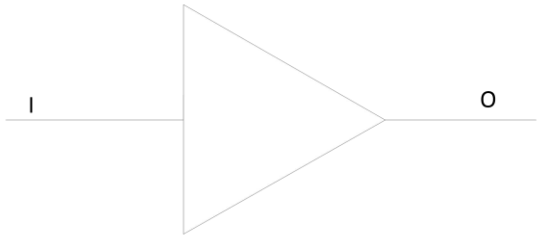


Figure 69 GTP_OUTBUF

3. Port description

Table 180 GTP_OUTBUF port list description

Port Name	Direction	Functional description
I	Input	Single-ended signal input
O	Output	Buffer output

4. Parameter description

Table 181 GTP_OUTBUF parameter description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 182	DEFAULT =LVCMOS33	Input IO standard
SLEW_RATE	"FAST", "SLOW", "MEDIUM"	"SLOW"	Slew rate
DRIVE_STRENGTH	2, 4, 6, 8, 12, 16, 24	"8"	Drive current intensity

Table 182 GTP_OUTBUF list of valid values for the parameter (Titan2)

GTP_OUTBUF		
IOSTANDARD	SLEW_RATE	DRIVE_STRENGTH
LVTTTL	FAST/SLOW	"4", "8", "12", "16", "24"
PCI33	None	None
LVCMOS33	FAST/SLOW	"4", "8", "12", "16"
LVCMOS25	FAST/SLOW	"4", "8", "12", "16"
LVCMOS18	FAST/SLOW/MEDIUM	"2", "4", "6", "8", "12", "16", "24" (HRIO)
LVCMOS15	FAST/SLOW/MEDIUM	"2", "4", "6", "8", "12", "16"
LVCMOS12	FAST/SLOW/MEDIUM	"2", "4", "6", "8"
SSTL18_I	FAST/SLOW/MEDIUM	None
SSTL18_II	FAST/SLOW/MEDIUM	None
SSTL15_I	FAST/SLOW/MEDIUM	None
SSTL15_II	FAST/SLOW/MEDIUM	None
SSTL135_I	FAST/SLOW/MEDIUM	None
SSTL135_II	FAST/SLOW/MEDIUM	None
SSTL12	FAST/SLOW/MEDIUM	None
HSTL18_I	FAST/SLOW/MEDIUM	None
HSTL18_II	FAST/SLOW/MEDIUM	None
HSTL15_I	FAST/SLOW/MEDIUM	None
HSTL15_II	FAST/SLOW/MEDIUM	None
HSTL12_I	FAST/SLOW/MEDIUM	None
HSUL12	FAST/SLOW/MEDIUM	None
POD12	FAST/SLOW/MEDIUM	None
SSTL18_I_CAL	FAST/SLOW/MEDIUM	None
SSTL18_II_CAL	FAST/SLOW/MEDIUM	None
SSTL15_I_CAL	FAST/SLOW/MEDIUM	None
SSTL15_II_CAL	FAST/SLOW/MEDIUM	None
SSTL135_I_CAL	FAST/SLOW/MEDIUM	None
SSTL135_II_CAL	FAST/SLOW/MEDIUM	None
SSTL12_CAL	FAST/SLOW/MEDIUM	None
HSTL18_I_CAL	FAST/SLOW/MEDIUM	None
HSTL18_II_CAL	FAST/SLOW/MEDIUM	None
HSTL15_I_CAL	FAST/SLOW/MEDIUM	None
HSTL15_II_CAL	FAST/SLOW/MEDIUM	None
HSTL12_I_CAL	FAST/SLOW/MEDIUM	None
HSUL12_CAL	FAST/SLOW/MEDIUM	None
POD12_CAL	FAST/SLOW/MEDIUM	None

GTP_OUTBUF		
IOSTANDARD	SLEW_RATE	DRIVE_STRENGTH
LVCAL_15	FAST/SLOW/MEDIUM	None
LVCAL_18	FAST/SLOW/MEDIUM	None
HSLVCAL_15	FAST/SLOW/MEDIUM	None
HSLVCAL_18	FAST/SLOW/MEDIUM	None

5. Instantiation template

```
GTP_OUTBUF#(
.IOSTANDARD ("DEFAULT"),
.SLEW_RATE ("SLOW"),
.DRIVE_STRENGTH (8)
)
GTP_OUTBUF_inst (
.I  (i),
.O  (o)
);
```

(8) GTP_OUTBUFT instruction

1. Supported devices

Table 183 GTP_OUTBUFT device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_OUTBUFT is a three-state output BUFFER. As shown in the figure below:

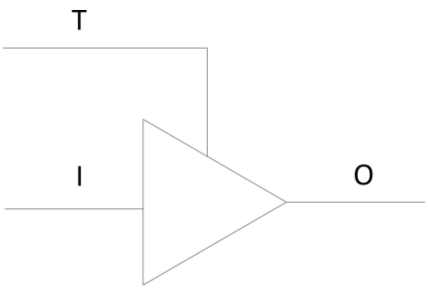


Figure 70 GTP_OUTBUFT

3. Port description

Table 184 GTP_OUTBUFT port list description

Port Name	Direction	Functional description
I	Input	Input signal
O	Output	Output signal
T	Input	Tri-state enable input, when T=1, the output tri-state (low level); when T=0, the input signal (I) from the chip is driven to the output

4. Parameter description

Table 185 GTP_OUTBUFT parameter description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 186	DEFAULT =LVCMOS33	Input IO standard
SLEW_RATE	"FAST", "SLOW", "MEDIUM"	"SLOW"	Slew rate
DRIVE_STRENGTH	"2", "4", "6", "8", "12", "16", "24"	"8"	Drive current intensity

Table 186 GTP_OUTBUFT list of valid values for the parameter (Titan2)

GTP_OUTBUFT		
IOSTANDARD	SLEW_RATE	DRIVE_STRENGTH
LVTTL33	FAST/SLOW	"4", "8", "12", "16", "24"
PCI33	None	None
LVCMOS33	FAST/SLOW	"4", "8", "12", "16"
LVCMOS25	FAST/SLOW	"4", "8", "12", "16"
LVCMOS18	FAST/SLOW/MEDIUM	"2", "4", "6", "8", "12", "16", "24" (HRIO)
LVCMOS15	FAST/SLOW/MEDIUM	"2", "4", "6", "8", "12", "16"
LVCMOS12	FAST/SLOW/MEDIUM	"2", "4", "6", "8"
SSTL18_I	FAST/SLOW/MEDIUM	None
SSTL18_II	FAST/SLOW/MEDIUM	None
SSTL15_I	FAST/SLOW/MEDIUM	None
SSTL15_II	FAST/SLOW/MEDIUM	None
SSTL135_I	FAST/SLOW/MEDIUM	None
SSTL135_II	FAST/SLOW/MEDIUM	None
SSTL12	FAST/SLOW/MEDIUM	None
HSTL18_I	FAST/SLOW/MEDIUM	None
HSTL18_II	FAST/SLOW/MEDIUM	None
HSTL15_I	FAST/SLOW/MEDIUM	None
HSTL15_II	FAST/SLOW/MEDIUM	None
HSTL12_I	FAST/SLOW/MEDIUM	None
HSUL12	FAST/SLOW/MEDIUM	None
POD12	FAST/SLOW/MEDIUM	None
SSTL18_I_CAL	FAST/SLOW/MEDIUM	None
SSTL18_II_CAL	FAST/SLOW/MEDIUM	None
SSTL15_I_CAL	FAST/SLOW/MEDIUM	None
SSTL15_II_CAL	FAST/SLOW/MEDIUM	None
SSTL135_I_CAL	FAST/SLOW/MEDIUM	None
SSTL135_II_CAL	FAST/SLOW/MEDIUM	None
SSTL12_CAL	FAST/SLOW/MEDIUM	None
HSTL18_I_CAL	FAST/SLOW/MEDIUM	None
HSTL18_II_CAL	FAST/SLOW/MEDIUM	None
HSTL15_I_CAL	FAST/SLOW/MEDIUM	None
HSTL15_II_CAL	FAST/SLOW/MEDIUM	None
HSTL12_I_CAL	FAST/SLOW/MEDIUM	None

GTP_OUTBUFT		
IOSTANDARD	SLEW_RATE	DRIVE_STRENGTH
HSUL12_CAL	FAST/SLOW/MEDIUM	None
POD12_CAL	FAST/SLOW/MEDIUM	None
LVCAL_15	FAST/SLOW/MEDIUM	None
LVCAL_18	FAST/SLOW/MEDIUM	None
HSLVCAL_15	FAST/SLOW/MEDIUM	None
HSLVCAL_18	FAST/SLOW/MEDIUM	None

5. Instantiation template

```
GTP_OUTBUFT#(
.IOSTANDARD ("DEFAULT"),
.SLEW_RATE ("SLOW "),
.DRIVE_STRENGTH (8)
)
GTP_OUTBUFT_inst (
.I   (i),
.O   (o),
.T   (t)
);
```

(9) GTP_OUTBUFDS instruction

1. Supported devices

Table 187 GTP_OUTBUFDS device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_OUTBUFDS is a true differential output buffer as shown below:

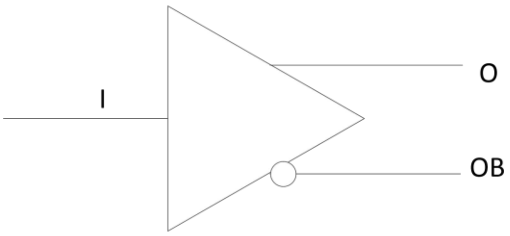


Figure 71 GTP_OUTBUFDS

3. Port description

Table 188 GTP_OUTBUFDS port list description

Port Name	Direction	Functional description
I	Input	Input signal
OB	Output	Differential N-terminal output signal
O	Output	Differential P terminal output signal

4. Parameter description

Table 189 GTP_OUTBUFDS parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	"LVDS", "MINI-LVDS", "SUB-LVDS", "TMDS", "RSDS", "PPDS", "LVDS_18"	DEFAULT= LVDS	Input IO standard

Note: The parameter IOSTANDARD setting value "LVDS_18" is dedicated to Titan2 HPIO, and the rest are HRIO value ranges.

5. Instantiation template

```
GTP_OUTBUFDS#(
.IOSTANDARD ("DEFAULT")
)
GTP_OUTBUFDS_inst (
.I  (i),
.OB (ob ),
.O  (o)
);
```

(10) GTP_OUTBUFTDS instruction

1. Supported devices

Table 190 GTP_OUTBUFTDS device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_OUTBUFTDS is a three-state output true differential BUFFER. As shown below:

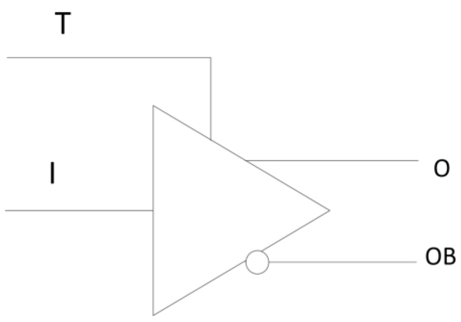


Figure 72 GTP_OUTBUFTDS

3. Port description

Table 191 GTP_OUTBUFTDS port list description

Port Name	Direction	Functional description
I	Input	Input signal
OB	Output	Differential N-terminal output signal

O	Output	Differential P terminal output signal
T	Input	Tri-state enable input, when T=1, the output tri-state (low level); when T=0, the input signal (I) from the chip is driven to the output

4. Parameter description

Table 192 GTP_OUTBUFTDS parameter description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	"LVDS", "MINI-LVDS", "SUB-LVDS", "TMDS", "RSDS", "PPDS", "LVDS_18"	DEFAULT=LVDS	Input IO standard

Note: The parameter IOSTANDARD setting value "LVDS_18" is dedicated to Titan2 HPIO, and the rest are HRIO value ranges.

5. Instantiation template

```
GTP_OUTBUFTDS#(
.IOSTANDARD ("DEFAULT")
)
GTP_OUTBUFTDS_inst (
.I   (i),
.T   (t),
.OB  (ob ),
.O   (o)
);
```

(11) GTP_OUTBUFCO instruction

1. Supported devices

Table 193 GTP_OUTBUFCO device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_OUTBUFCO is a pseudo-differential output buffer. As shown below:

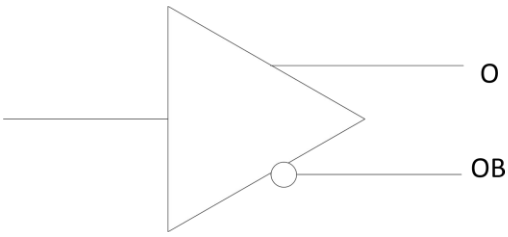


Figure 73 GTP_OUTBUFCO

3. Port description

Table 194 GTP_OUTBUFCO port list information

Port Name	Direction	Functional description
I	Input	Input signal
OB	Output	Differential N output signal

O	Output	Differential P output signal

4. Parameter description

Table 195 GTP_OUTBUFCO parameter list information

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 196	DEFAULT=LVCMOS33	Input IO standard

Table 196 GTP_OUTBUFCO list of valid values for the parameter (Titan2)

GTP_OUTBUFCO
IOSTANDARD
SSTL18D_I
SSTL18D_II
SSTL15D_I
SSTL15D_II
SSTL135D_I
SSTL135D_II
SSTL12D
HSTL15D_I
HSTL15D_II
HSTL18D_I
HSTL18D_II
HSTL12D_I
HSUL12D
POD12D
SSTL18D_I_CAL
SSTL18D_II_CAL
SSTL15D_I_CAL
SSTL15D_II_CAL
SSTL135D_I_CAL
SSTL135D_II_CAL
SSTL12D_CAL
HSTL15D_I_CAL
HSTL15D_II_CAL
HSTL18D_I_CAL
HSTL18D_II_CAL
HSTL12D_I_CAL
HSUL12D_CAL
POD12D_CAL

5. Instantiation template

```
GTP_OUTBUFCO#(
.IOSTANDARD ("LVCMOS33")
)
```

```
GTP_OUTBUFCO_inst (
.I  (i),
.OB (ob ),
.O  (o)
);
```

(12) GTP_OUTBUFTCO instruction

1. Supported devices

Table 197 GTP_OUTBUFTCO device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_OUTBUFTCO is a three-state output pseudo-differential BUFFER. As shown in the figure below:

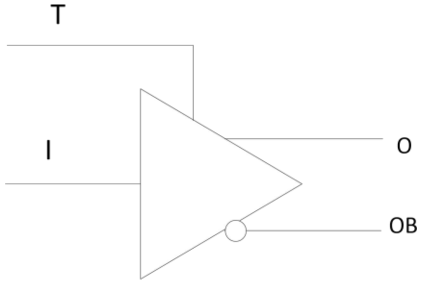


Figure 74 GTP_OUTBUFTCO

3. Port description

Table 198 GTP_OUTBUFTCO port list description

Port Name	Direction	Functional description
I	Input	Input signal
OB	Output	Differential N-terminal output signal
O	Output	Differential P terminal output signal
T	Input	Tri-state enable input, when T=1, the output tri-state (low level); when T=0, the input signal (I) from the chip is driven to the output

4. Parameter description

Table 199 GTP_OUTBUFTCO parameter description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 200	DEFAULT =LVCMOS33	Input IO standard

Table 200 GTP_OUTBUFTCO list of valid values for the parameter (Titan2)

GTP_OUTBUFTCO
IOSTANDARD

GTP_OUTBUFTCO
IOSTANDARD
SSTL18D_I
SSTL18D_II
SSTL15D_I
SSTL15D_II
SSTL135D_I
SSTL135D_II
SSTL12D
HSTL15D_I
HSTL15D_II
HSTL18D_I
HSTL18D_II
HSTL12D_I
HSUL12D
POD12D
SSTL18D_I_CAL
SSTL18D_II_CAL
SSTL15D_I_CAL
SSTL15D_II_CAL
SSTL135D_I_CAL
SSTL135D_II_CAL
SSTL12D_CAL
HSTL15D_I_CAL
HSTL15D_II_CAL
HSTL18D_I_CAL
HSTL18D_II_CAL
HSTL12D_I_CAL
HSUL12D_CAL
POD12D_CAL

5. Instantiation template

```
GTP_OUTBUFTCO#(
.IOSTANDARD ("DEFAULT")
)
GTP_OUTBUFTCO_inst (
.I  (i),
.T  (t),
.OB (ob ),
.O  (o)
);
```

(13) GTP_IOBUF instruction

1. Supported devices

Table 201 GTP_IOBUF device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IOBUF is a bidirectional single-signal BUFFER, which connects external bidirectional pins and internal logic circuits. As shown in the figure below:

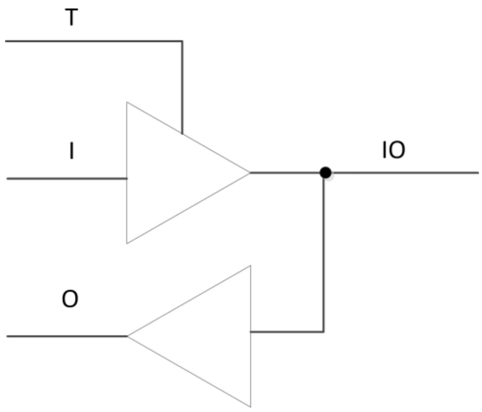


Figure 75 GTP_IOBUF

3. Port description

Table 202 GTP_IOBUF port list description

Port Name	Direction	Functional description
I	Input	Input signal
T	Input	Three-state input enable, active low
O	Output	Output signal
IO	input Output	Input and output signals (inout)

4. Parameter description

Table 203 GTP_IOBUF parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 204	DEFAULT =LVCMOS33	Input IO standard
TERM_DDR	"ON", "OFF"	"ON"	HSTL, SSTL standard input, built-in terminal resistor on or off
SLEW_RATE	"SLOW", "FAST", "MEDIUM"	"SLOW"	Slew rate
DRIVE_STRENGTH	"2", "4", "6", "8", "12", "16", "24"	"8"	Drive current intensity

Table 204 GTP_IOBUF list of valid values for the parameter (Titan2)

GTP_IOBUF

IOSTANDARD	SLEW_RATE	TERM_DDR	DRIVE_STRENGTH
LVTTL33	FAST/SLOW	None	"4", "8", "12", "16", "24"
PCI33	None	None	None
LVCMOS33	FAST/SLOW	None	"4", "8", "12", "16"
LVCMOS25	FAST/SLOW	None	"4", "8", "12", "16"
LVCMOS18	FAST/SLOW/MEDIUM	None	"2", "4", "6", "8", "12", "16", "24" (HRIO)
LVCMOS15	FAST/SLOW/MEDIUM	None	"2", "4", "6", "8", "12", "16"
LVCMOS12	FAST/SLOW/MEDIUM	None	"2", "4", "6", "8"
SSTL18_II	FAST/SLOW/MEDIUM	OFF/ON	None
SSTL15_II	FAST/SLOW/MEDIUM	OFF/ON	None
SSTL135_II	FAST/SLOW/MEDIUM	OFF/ON	None
SSTL12	FAST/SLOW/MEDIUM	OFF/ON	None
HSTL18_II	FAST/SLOW/MEDIUM	OFF/ON	None
HSTL15_II	FAST/SLOW/MEDIUM	OFF/ON	None
HSUL12	FAST/SLOW/MEDIUM	OFF/ON	None
POD12	FAST/SLOW/MEDIUM	OFF	None
SSTL18_II_CAL	FAST/SLOW/MEDIUM	ON	None
SSTL15_II_CAL	FAST/SLOW/MEDIUM	ON	None
SSTL135_II_CAL	FAST/SLOW/MEDIUM	ON	None
SSTL12_CAL	FAST/SLOW/MEDIUM	ON	None
HSTL18_II_CAL	FAST/SLOW/MEDIUM	ON	None
HSTL15_II_CAL	FAST/SLOW/MEDIUM	ON	None
HSUL12_CAL	FAST/SLOW/MEDIUM	ON	None
POD12_CAL	FAST/SLOW/MEDIUM	ON	None
LVCAL_15	FAST/SLOW/MEDIUM	ON	None
LVCAL_18	FAST/SLOW/MEDIUM	ON	None
HSLVCAL_15	FAST/SLOW/MEDIUM	ON	None
HSLVCAL_18	FAST/SLOW/MEDIUM	ON	None

5. Instantiation template

```
GTP_IOBUF#(
.IOSTANDARD ("DEFAULT"),
.SLEW_RATE ("SLOW"),
.DRIVE_STRENGTH ("8"),
.TERM_DDR("ON")
)
GTP_IOBUF_inst (
.I  (i),
.T  (t),
.IO (io ),
.O  (o)
);
```

(14) GTP_IOBUFCO instruction

1. Supported devices

Table 205 GTP_IOBUFCO device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IOBUFCO is a bidirectional pseudo-differential buffer, which only supports pseudo-differential levels (corresponding to GTP_IOBUFDS, GTP_IOBUFDS only supports true differential levels). Its structural block diagram is as follows:

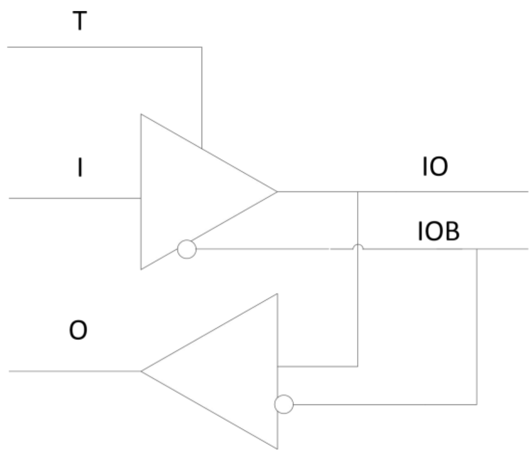


Figure 76 GTP_IOBUFCO

3. Port description

Table 206 GTP_IOBUFCO port list description

Port Name	Direction	Functional description
I	Input	Input signal
T	Input	Tri-state enable input, when T=1, the output tri-state (low level); when T=0, the input signal (I) from the chip is driven to the output
IO	Input / Output	Differential P-side input/output
IOB	Input / Output	Differential Q-terminal input/output
O	Output	Output signal

4. Parameter description

Table 207 GTP_IOBUFCO parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 208, form 209	DEFAULT =LVCMOS33	Input IO standard
TERM_DDR	"ON", "OFF"	"ON"	HSTL, SSTL standard input, built-in terminal resistor on or off

Table 208 GTP_IOBUFCO list of valid values for the parameter (Titan2 HPIO)

GTP_IOBUFCO（Titan2 HPIO）	
IOSTANDARD	TERM_DDR
SSTL18D_I	OFF/ON
SSTL18D_II	OFF/ON
SSTL15D_I	OFF/ON
SSTL15D_II	OFF/ON
SSTL135D_I	OFF/ON
SSTL135D_II	OFF/ON
SSTL12D	OFF/ON
HSTL12D_I	OFF/ON
HSTL15D_I	OFF/ON
HSTL15D_II	OFF/ON
HSTL18D_I	OFF/ON
HSTL18D_II	OFF/ON
HSUL12D	OFF/ON
POD12D	OFF/ON
SSTL18D_I_CAL	ON
SSTL18D_II_CAL	ON
SSTL15D_I_CAL	ON
SSTL15D_II_CAL	ON
SSTL135D_I_CAL	ON
SSTL135D_II_CAL	ON
SSTL12D_CAL	ON
HSTL12D_I_CAL	ON
HSTL15D_I_CAL	ON
HSTL15D_II_CAL	ON
HSTL18D_I_CAL	ON
HSTL18D_II_CAL	ON
HSUL12D_CAL	ON
POD12D_CAL	ON

Table 209 GTP_IOBUFCO list of valid values for the parameter (Titan2 HRIO)

GTP_IOBUFCO（Titan2 HRIO）	
IOSTANDARD	TERM_DDR
SSTL18D_I	OFF/ON
SSTL18D_II	OFF/ON
SSTL15D_I	OFF/ON
SSTL15D_II	OFF/ON
SSTL135D_I	OFF/ON
SSTL135D_II	OFF/ON
HSTL15D_I	OFF/ON
HSTL15D_II	OFF/ON
HSTL18D_I	OFF/ON
HSTL18D_II	OFF/ON
HSUL12D	OFF/ON

5. Instantiation template

```
GTP_IOBUFCO#(
.IOSTANDARD ("DEFAULT"),
.TERM_DDR  ("ON")
)
GTP_IOBUFCO_inst(
.I   (i),
.T   (t),
.IO  (io),
.IOB(iob),
.O   (o)
);
```

(15) GTP_IOBUFDS instruction

1. Supported devices

Table 210 GTP_IOBUFDS device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IOBUFDS is a bidirectional single-signal differential input BUFFER. As shown in the figure below:

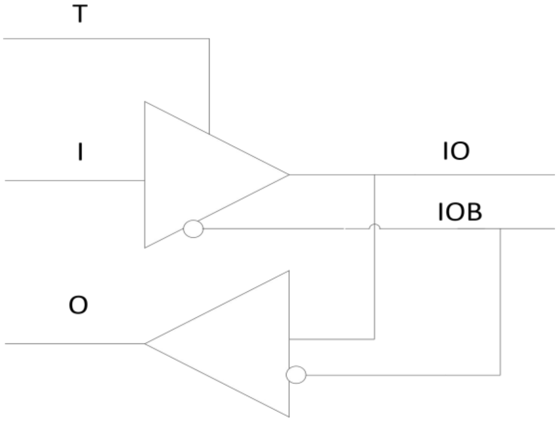


Figure 77 GTP_IOBUFDS

3. Port description

Table 211 GTP_IOBUFDS port list description

Port Name	Direction	Functional description
I	Input	Input signal
T	Input	Three-state input enable, active low
O	Output	Output signal

IO	input Output	Differential P input and output (inout)
IOB	input Output	Differential N input and output

4. Parameter description

Table 212 GTP_IOBUFDS parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	"LVDS", "MINI-LVDS", "PPDS", "RSDS", "SUB-LVDS", "TMDS", "LVDS_18"	DEFAULT=LVDS	Input IO standard
TERM_DIFF	"ON", "OFF"	"ON"	For standard input such as LVDS, the built-in differential terminal resistor can be turned on or off

Note: The parameter IOSTANDARD setting value "LVDS_18" is dedicated to Titan2 HPIO, and the rest are HRIO value ranges.

5. Instantiation template

```
GTP_IOBUFDS#(
.IOSTANDARD ("DEFAULT"),
.TERM_DIFF("ON")
)
GTP_IOBUFDS_inst (
.I            (i),
.T            (t),
.IO           (io),
.IOB          (iob),
.O            (o)
);
```

(16) GTP_IOBUF_RX_MIPI instruction

1. Supported devices

Table 213 GTP_IOBUF_RX_MIPI device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IOBUF_RX_MIPI supports MIPI DPHY high-speed input, and single-ended input and output in low power mode (LP).

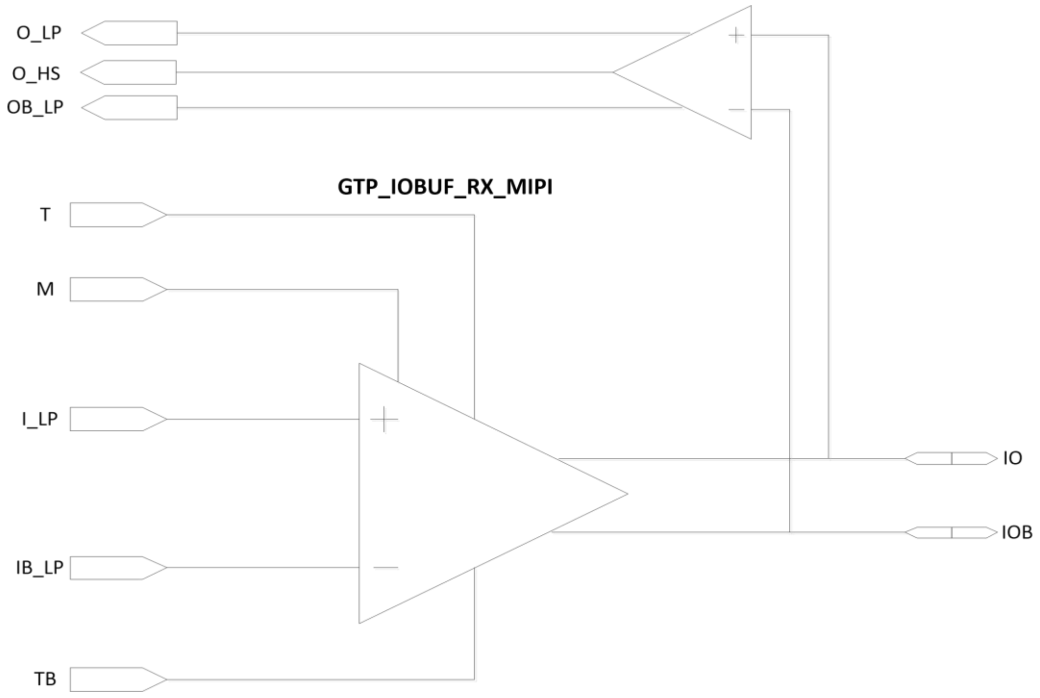


Figure 78 GTP_IOBUF_RX_MIPI

3. Port description

Table 214 GTP_IOBUF_RX_MIPI port list description

Port Name	Direction	Functional description
I_LP	Input	Input signal of single-ended output buffer in LP mode
IB_LP	Input	Input signal of single-ended output buffer in LP mode
M	Input	Mode selection signal. 1: HS mode, differential input; 0: LP mode
T	Input	Mode selection signal. 1: HS mode, differential input; 0: LP mode
TB	Input	Single-ended output enable signal
O_HS	Output	Single-ended output enable signal
O_LP	Output	Single-ended output enable signal
OB_LP	Output	Single-ended output enable signal
IO	Two-way	Non-inverting differential input (HS) and input/output (LP)
IOB	Two-way	Inverting differential input (HS) and input/output (LP)

4. Parameter description

Table 215 GTP_IOBUF_RX_MIPI parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	"MIPI"	"DEFAULT"	Input IO standard
DRIVE_STRENGTH	"2", "6"	"6"	Drive current intensity
SLEW_RATE	"SLOW", "FAST"	"SLOW"	Slew rate
TERM_DIFF	"ON", "OFF"	"OFF"	For differential input, the built-in terminal resistor is turned on or off
HPIO	"TRUE", "FALSE"	"FALSE"	Mapped to HPIO BANK is "TRUE"; Mapped to HRIO BANK is "FALSE";

5. Instantiation template

```
GTP_IOBUF_RX_MIPI # (
.IOSTANDARD   ( "DEFAULT" ),//"MIPI", "DEFAULT"
.SLEW_RATE    ("SLOW"),
.DRIVE_STRENGTH("6"),
.TERM_DIFF    ( "ON" ),
.HPIO         ( "FALSE" )
)
GTP_IOBUF_RX_MIPI _inst (
.O_LP         ( O_LP    ),
.OB_LP        ( OB_LP   ),
.O_HS         ( O_HS    ),
.IO           ( IO      ),
.IOB          ( IOB     ),
.I_LP         ( I_LP    ),
.IB_LP        ( IB_LP   ),
.T            ( T       ),
.TB           ( TB      ),
.M            ( M       )
);
```

(17) GTP_IOBUF_TX_MIPI instruction

1. Supported devices

Table 216 GTP_IOBUF_TX_MIPI device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IOBUF_TX_MIPI supports MIPI DPHY tri-state output, including input/output in low-power mode (LP) and differential output in high-speed (HS) mode, and the two modes can be switched freely according to actual applications.

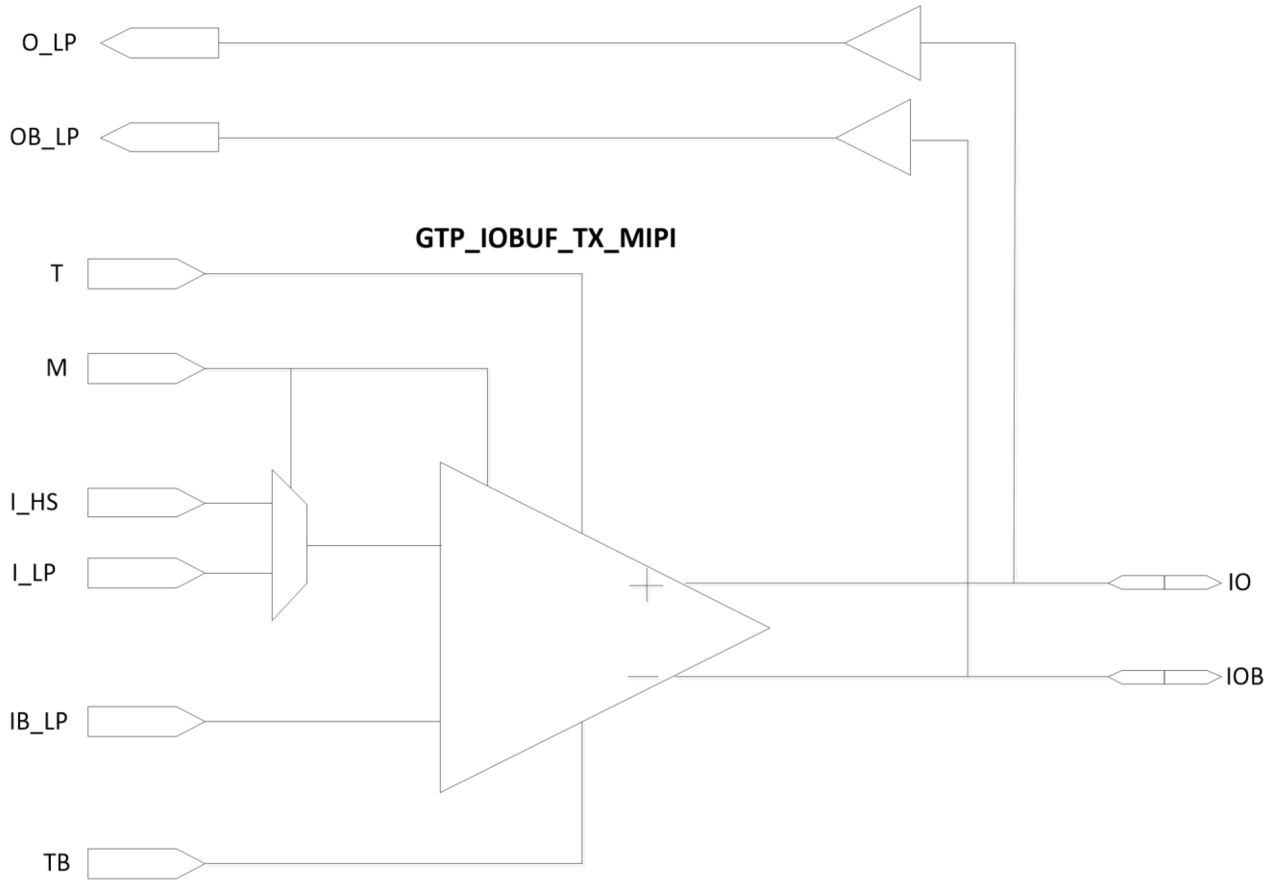


Figure 79 GTP_IOBUF_TX_MIPI

3. Port description

Table 217 GTP_IOBUF_TX_MIPI port list description

Port Name	Direction	Functional description
I_HS	Input	HS Differential output buffer input signal, from IOL
I_LP	Input	Input signal of single-ended output buffer in LP mode, from IOL
IB_LP	Input	Input signal of single-ended output buffer in LP mode, from IOL
M	Input	Mode selection signal. 1: HS mode, differential output; 0: LP mode, single-ended input and output. From IOL
T	Input	Differential and single-ended output enable signals
TB	输入	Single-ended output enable signal
O_LP	Output	Single-Ended (LP) Input to IOL
OB_LP	Output	Single-Ended (LP) Input to IOL
IO	Two-way	In-phase differential output
IOB	Two-way	Reverse differential output

4. Parameter description

Table 218 GTP_IOBUF_TX_MIPI parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	MIPI	DEFAULT	Input IO standard VDDIO=1.2V
DRIVE_STRENGTH	"2", "6"	"6"	Drive current intensity

SLEW_RATE	"SLOW", "FAST"	"SLOW"	Slew rate
TERM_DIFF	"ON", "OFF"	"ON"	For differential input, the built-in terminal resistor is turned on or off

5. Instantiation template

```
GTP_IOBUF_TX_MIPI # (
.IOSTANDARD   ( "DEFAULT" ),//"MIPI", "DEFAULT"
.SLEW_RATE("SLOW"),
.DRIVE_STRENGTH("6"),
.TERM_DIFF    ( "ON"    )
)
GTP_IOBUF_TX_MIPI _inst (
.O_LP         ( O_LP    ),
.OB_LP        ( OB_LP   ),
.IO           ( IO      ),
.IOB          ( IOB     ),
.I_LP         ( I_LP    ),
.I_HS         ( I_HS    ),
.IB_LP        ( IB_LP   ),
.T            ( T       ),
.TB           ( TB      ),
.M            ( M       )
);
```

(18) GTP_IOBUFE instruction

1. Supported devices

Table 219 GTP_IOBUFE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IOBUFE supports single-ended input and output drive functions. As shown in the figure below:

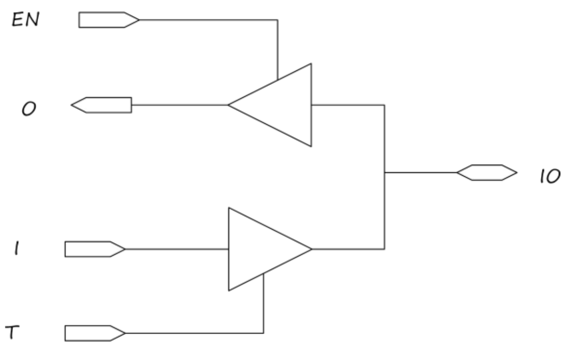


Figure 80 GTP_IOBUFE

3. Port description

Table 220 GTP_IOBUFE port list description

Port Name	Direction	Functional description
I	Input	Single-ended signal input
O	Output	The output of the input buffer is sent to the chip
T	Input	Enable output buffer
IO	Two-way	PAD
EN	Input	Turn off the input buffer when it is 0.

4. Parameter description

Table 221 GTP_IOBUFE parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 222	DEFAULT =LVCMOS33	Input IO standard
TERM_DDR	"ON", "OFF"	"ON"	HSTL, SSTL standard input, built-in terminal resistor on or off
SLEW_RATE	"SLOW", "FAST", "MEDIUM"	"SLOW"	Slew rate
DRIVE_STRENGTH	"2", "4", "6", "8", "12", "16", "24"	"8"	Drive current intensity

Table 222 GTP_IOBUFE list of valid values for the parameter (Titan2)

GTP_IOBUFE			
IOSTANDARD	SLEW_RATE	TERM_DDR	DRIVE_STRENGTH
LVTTTL	FAST/SLOW	None	"4", "8", "12", "16", "24"
PCI33	None	None	None
LVCMOS33	FAST/SLOW	None	"4", "8", "12", "16"
LVCMOS25	FAST/SLOW	None	"4", "8", "12", "16"
LVCMOS18	FAST/SLOW/MEDIUM	None	"2", "4", "6", "8", "12", "16", "24" (HRIO)
LVCMOS15	FAST/SLOW/MEDIUM	None	"2", "4", "6", "8", "12", "16"
LVCMOS12	FAST/SLOW/MEDIUM	None	"2", "4", "6", "8"
SSTL18_I	FAST/SLOW/MEDIUM	OFF/ON	None
SSTL18_II	FAST/SLOW/MEDIUM	OFF/ON	None

GTP_IOBUFE			
IOSTANDARD	SLEW_RATE	TERM_DDR	DRIVE_STRENGTH
SSTL15_I	FAST/SLOW/MEDIUM	OFF/ON	None
SSTL15_II	FAST/SLOW/MEDIUM	OFF/ON	None
SSTL135_I	FAST/SLOW/MEDIUM	OFF/ON	None
SSTL135_II	FAST/SLOW/MEDIUM	OFF/ON	None
SSTL12	FAST/SLOW/MEDIUM	OFF/ON	None
HSTL18_I	FAST/SLOW/MEDIUM	OFF/ON	None
HSTL18_II	FAST/SLOW/MEDIUM	OFF/ON	None
HSTL15_I	FAST/SLOW/MEDIUM	OFF/ON	None
HSTL15_II	FAST/SLOW/MEDIUM	OFF/ON	None
HSTL12_I	FAST/SLOW/MEDIUM	OFF/ON	None
HSUL12	FAST/SLOW/MEDIUM	OFF/ON	None
POD12	FAST/SLOW/MEDIUM	OFF/ON	None
SSTL18_I_CAL	FAST/SLOW/MEDIUM	ON	None
SSTL18_II_CAL	FAST/SLOW/MEDIUM	ON	None
SSTL15_I_CAL	FAST/SLOW/MEDIUM	ON	None
SSTL15_II_CAL	FAST/SLOW/MEDIUM	ON	None
SSTL135_I_CAL	FAST/SLOW/MEDIUM	ON	None
SSTL135_II_CAL	FAST/SLOW/MEDIUM	ON	None
SSTL12_CAL	FAST/SLOW/MEDIUM	ON	None
HSTL18_I_CAL	FAST/SLOW/MEDIUM	ON	None
HSTL18_II_CAL	FAST/SLOW/MEDIUM	ON	None
HSTL15_I_CAL	FAST/SLOW/MEDIUM	ON	None
HSTL15_II_CAL	FAST/SLOW/MEDIUM	ON	None
HSTL12_I_CAL	FAST/SLOW/MEDIUM	ON	None
HSUL12_CAL	FAST/SLOW/MEDIUM	ON	None
POD12_CAL	FAST/SLOW/MEDIUM	ON	None
LVCAL_15	FAST/SLOW/MEDIUM	ON	None
LVCAL_18	FAST/SLOW/MEDIUM	ON	None
HSLVCAL_15	FAST/SLOW/MEDIUM	ON	None
HSLVCAL_18	FAST/SLOW/MEDIUM	ON	None

5. Instantiation template

```
GTP_IOBUFE #(
.IOSTANDARD("DEFAULT"),
.SLEW_RATE("SLOW"),
.DRIVE_STRENGTH("8"),
.TERM_DDR("ON")
) GTP_IOBUFE_inst (
.IO          (io),
.O           (o ),
.EN          (en),
.I           (i ),
.T           (t )
);
```

(19) GTP_IOBUFECO instruction

1. Supported devices

Table 223 GTP_IOBUFECO device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IOBUFECO supports single-ended input and pseudo-differential output drive function. As shown below:

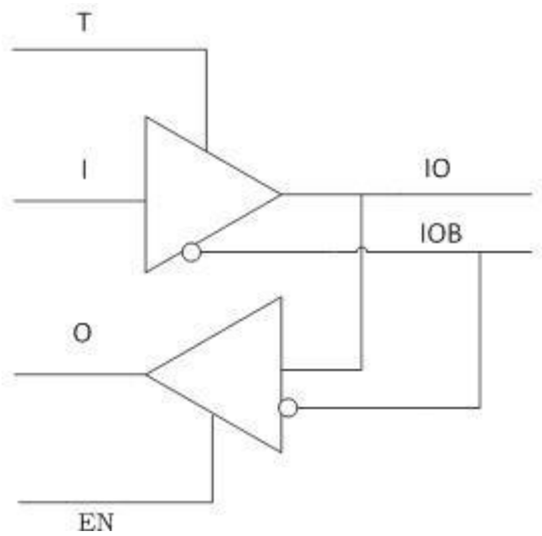


Figure 81 GTP_IOBUFECO

3. Port description

Table 224 GTP_IOBUFECO port list description

Port Name	Direction	Functional description
I	Input	Single-ended signal input
O	Output	Input buffer output, fed into the chip internal
T	Input	Enable output buffer
IO	Two-way	First IO
IOB	Two-way	The second IO, its value is the inverse of IO
EN	Input	Turn off the input buffer when it is 0.

4. Parameter description

Table 225 GTP_IOBUFECO parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 226, table 227	DEFAULT =LVCMOS33	Input IO standard
TERM_DDR	"ON", "OFF"	"ON"	HSTL, SSTL standard input, built-in terminal resistor on or off

Table 226 GTP_IOBUFECO list of valid values for the parameter (Titan2 HPIO)

GTP_IOBUFECO (Titan2 HPIO)	
IOSTANDARD	TERM_DDR
SSTL18D_I	OFF/ON
SSTL18D_II	OFF/ON
SSTL15D_I	OFF/ON
SSTL15D_II	OFF/ON
SSTL135D_I	OFF/ON
SSTL135D_II	OFF/ON
SSTL12D	OFF/ON
HSTL12D_I	OFF/ON
HSTL15D_I	OFF/ON
HSTL15D_II	OFF/ON
HSTL18D_I	OFF/ON
HSTL18D_II	OFF/ON
HSUL12D	OFF/ON
POD12D	OFF/ON
SSTL18D_I_CAL	ON
SSTL18D_II_CAL	ON
SSTL15D_I_CAL	ON
SSTL15D_II_CAL	ON
SSTL135D_I_CAL	ON
SSTL135D_II_CAL	ON
SSTL12D_CAL	ON
HSTL12D_I_CAL	ON
HSTL15D_I_CAL	ON
HSTL15D_II_CAL	ON
HSTL18D_I_CAL	ON
HSTL18D_II_CAL	ON
HSUL12D_CAL	ON
POD12D_CAL	ON

Table 227 GTP_IOBUFECO list of valid values for the parameter (Titan2 HRIO)

GTP_IOBUFECO (Titan2 HRIO)	
IOSTANDARD	TERM_DDR
SSTL18D_I	OFF/ON
SSTL18D_II	OFF/ON
SSTL15D_I	OFF/ON
SSTL15D_II	OFF/ON
SSTL135D_I	OFF/ON
SSTL135D_II	OFF/ON
HSTL15D_I	OFF/ON
HSTL15D_II	OFF/ON
HSTL18D_I	OFF/ON
HSTL18D_II	OFF/ON
HSUL12D	OFF/ON

5. Instantiation template

```
GTP_IOBUFECO #(
.IOSTANDARD("DEFAULT"),
.TERM_DDR("ON")
) GTP_IOBUFECO_inst (
.IO            (io ),
.IOB           (iob),
.O             (o  ),
.EN            (en ),
.I             (i  ),
.T             (t  )
);
```

(20) GTP_IOBUFEDS instruction

1. Supported devices

Table 228 GTP_IOBUFEDS device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IOBUFEDS is a true differential Buffer. As shown in the figure below, the differential output under LVDS and other standards (input is I, output is IO/IOB), and at the same time, it can be configured as an input (IO, IOB > O path in the figure)

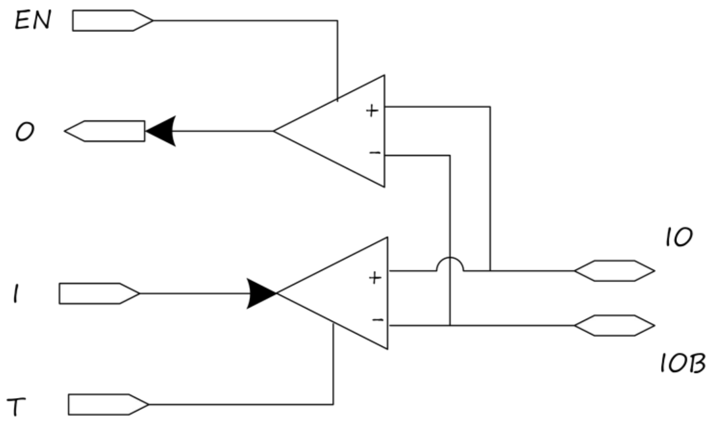


Figure 82 GTP_IOBUFEDS

3. Port description

Table 229 GTP_IOBUFEDS port list description

Port Name	Direction	Functional description
I	Input	Single-ended signal input
O	Output	Input buffer output, fed into the chip internal
T	Input	Enable signal
IO	Two-way	IO PAD

IOB	Two-way	IO's PAD, opposite to the value of IO
EN	Input	Turn off the input buffer when it is 0.

4. Parameter description

Table 230 GTP_IOBUFEDS parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	"LVDS", "MINI-LVDS", "PPDS", "RSDS", "SUB-LVDS", "TMDS", "LVDS_18"	DEFAULT=LVDS	Input IO standard
TERM_DIFF	"ON", "OFF"	"ON"	For standard input such as LVDS, the built-in differential terminal resistor can be turned on or off

Note: The parameter IOSTANDARD setting value "LVDS_18" is dedicated to Titan2 HPIO, and the rest are HRIO value ranges..

5. Instantiation template

```
GTP_IOBUFEDS #(
.IOSTANDARD("DEFAULT"),
.TERM_DIFF("ON")
) GTP_IOBUFEDS_inst (

.IO            (io        ),
.IOB           (iob       ),
.O             (o         ),
.EN            (en        ),
.I             (I         ),
.T             (t         )
);
```

(21) GTP_IOBUFCO_E1 instruction

1. Supported devices

Table 231 GTP_IOBUFCO_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IOBUFCO_E1 supports differential input, pseudo-differential (two single-ended implementation) output drive function. Its structural block diagram is as follows:

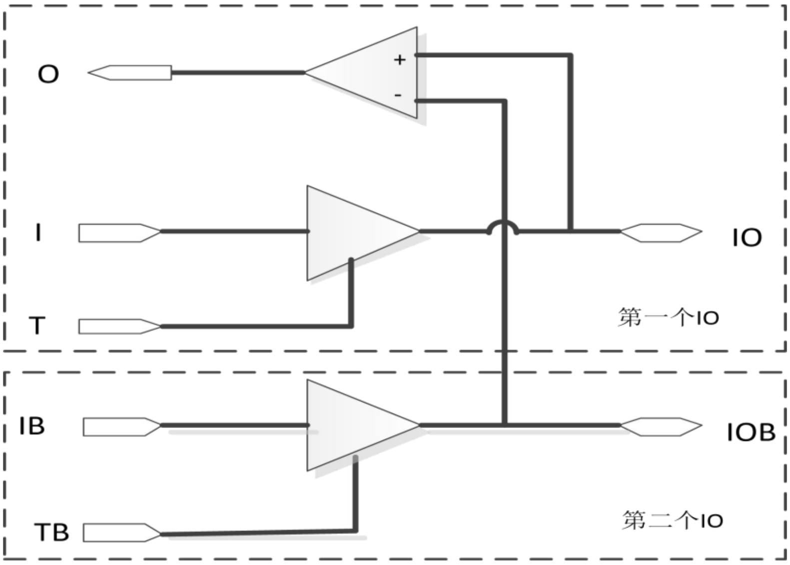


Figure 83 GTP_IOBUFCO_E1

3. Port description

Table 232 GTP_IOBUFCO_E1 port list description

Port Name	Direction	Functional description
I	IN	The first IO single-ended signal input
IB	IN	The second IO single-ended signal input
O	OUT	Input buffer output, fed into the chip internal
T	IN	The second IO single-ended signal input
TB	IN	The second IO enables output buffer
IO	INOUT	The second IO enables output buffer
IOB	INOUT	The PAD of the second IO, its value is the inverse of IO

4. Parameter description

Table 233 GTP_IOBUFCO_E1 parameter list description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 234, table 235	DEFAULT = POD12D	Input IO standard
TERM_DDR	"ON", "OFF"	"ON"	HSTL, SSTL standard input, built-in terminal resistor on or off

Table 234 GTP_IOBUFCO_E1 list of valid values for the parameter (Titan2 HPIO)

GTP_IOBUFCO_E1（Titan2 HPIO）	
IOSTANDARD	TERM_DDR
SSTL18D_I	OFF/ON
SSTL18D_II	OFF/ON
SSTL15D_I	OFF/ON
SSTL15D_II	OFF/ON
SSTL135D_I	OFF/ON

GTP_IOBUFCO_E1 (Titan2 HPIO)	
IOSTANDARD	TERM_DDR
SSTL135D_II	OFF/ON
SSTL12D	OFF/ON
HSTL12D_I	OFF/ON
HSTL15D_I	OFF/ON
HSTL15D_II	OFF/ON
HSTL18D_I	OFF/ON
HSTL18D_II	OFF/ON
HSUL12D	OFF/ON
POD12D	OFF/ON
SSTL18D_I_CAL	ON
SSTL18D_II_CAL	ON
SSTL15D_I_CAL	ON
SSTL15D_II_CAL	ON
SSTL135D_I_CAL	ON
SSTL135D_II_CAL	ON
SSTL12D_CAL	ON
HSTL12D_I_CAL	ON
HSTL15D_I_CAL	ON
HSTL15D_II_CAL	ON
HSTL18D_I_CAL	ON
HSTL18D_II_CAL	ON
HSUL12D_CAL	ON
POD12D_CAL	ON

Table 235 GTP_IOBUFCO_E1 list of valid values for the parameter (Titan2 HRIO)

GTP_IOBUFCO_E1（Titan2 HRIO）	
IOSTANDARD	TERM_DDR
SSTL18D_I	OFF/ON
SSTL18D_II	OFF/ON
SSTL15D_I	OFF/ON
SSTL15D_II	OFF/ON
SSTL135D_I	OFF/ON
SSTL135D_II	OFF/ON
HSTL15D_I	OFF/ON
HSTL15D_II	OFF/ON
HSTL18D_I	OFF/ON
HSTL18D_II	OFF/ON
HSUL12D	OFF/ON

5. Instantiation template

```
GTP_IOBUFCO_E1 #(
.IOSTANDARD("DEFAULT"),
.TERM_DDR("ON")
```

```
) GTP_IOBUFCO_E1_inst (
.IO(), // INOUT
.IOB(),// INOUT
.O(),  // OUTPUT
.I(),  // INPUT
.IB(), // INPUT
.T(),  // INPUT
.TB()  // INPUT
);
```

(22) GTP_OUTBUFCO_E1 instruction

1. Supported devices

Table 236 GTP_OUTBUFCO_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_OUTBUFCO_E1 supports pseudo-differential (two single-ended implementation) output drive function. As shown below:

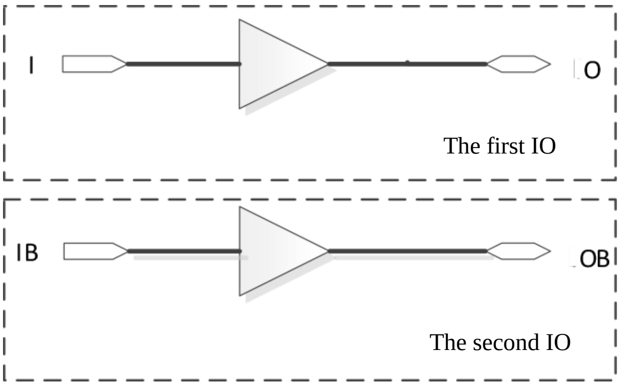


Figure 84 GTP_OUTBUFCO_E1

3. Port description

Table 237 GTP_OUTBUFCO_E1 port list information

Port Name	Direction	Functional description
I	IN	The first IO single-ended signal input
IB	IN	The second IO single-ended signal input, its value is the inverse of I
O	OUT	PAD of the first IO
OB	OUT	The PAD of the second IO, its value is the inverse of IO

4. Parameter description

Table 238 GTP_OUTBUFCO_E1 parameter list information

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 239, table 240	DEFAULT = POD12D	Input IO standard

Table 239 GTP_OUTBUFCO_E1 list of valid values for the parameter (Titan2 HPIO)

GTP_OUTBUFCO_E1
IOSTANDARD
SSTL18D_I
SSTL18D_II
SSTL15D_I
SSTL15D_II
SSTL135D_I
SSTL135D_II
SSTL12D
HSTL12D_I
HSTL15D_I
HSTL15D_II
HSTL18D_I
HSTL18D_II
HSUL12D
POD12D
SSTL18D_I_CAL
SSTL18D_II_CAL
SSTL15D_I_CAL
SSTL15D_II_CAL
SSTL135D_I_CAL
SSTL135D_II_CAL
SSTL12D_CAL
HSTL12D_I_CAL
HSTL15D_I_CAL
HSTL15D_II_CAL
HSTL18D_I_CAL
HSTL18D_II_CAL
HSUL12D_CAL
POD12D_CAL

Table 240 GTP_OUTBUFCO_E1 list of valid values for the parameter (Titan2 HRIO)

GTP_OUTBUFCO_E1
IOSTANDARD
SSTL18D_I
SSTL18D_II
SSTL15D_I
SSTL15D_II
SSTL135D_I
SSTL135D_II
HSTL15D_I
HSTL15D_II
HSTL18D_I
HSTL18D_II
HSUL12D

5. Instantiation template

```
GTP_OUTBUFCO_E1 #(
.IOSTANDARD("DEFAULT")
) GTP_OUTBUFCO_E1_inst (
.O(), // OUTPUT
.OB(),// OUTPUT
.I(), // INPUT
.IB() // INPUT
);
```

(23) GTP_OUTBUFTCO_E1 instruction

1. Supported devices

Table 241 GTP_OUTBUFTCO_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_OUTBUFTCO_E1 supports pseudo-differential (two single-ended implementation) output drive function. As shown below:

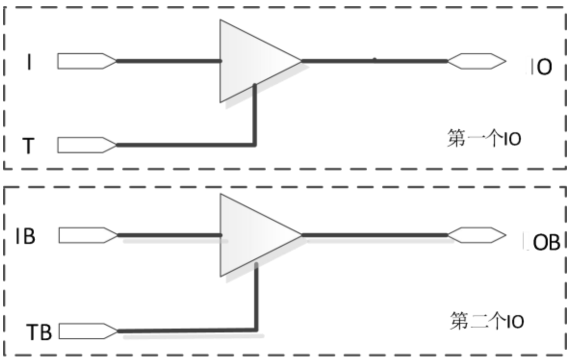


Figure 85 GTP_OUTBUFTCO_E1

3. Port description

Table 242 GTP_OUTBUFTCO_E1 port list description

Port Name	Direction	Functional description
I	IN	The first IO single-ended signal input
B	IN	The second IO single-ended signal input, its value is the inverse of I
T	IN	The first IO enable output buffer
TB	IN	The second IO enables output buffer
O	OUT	PAD of the first IO

4. Parameter description

Table 243 GTP_OUTBUFTCO_E1 parameter description

Parameter Name	Settings	Defaults	Functional description
IOSTANDARD	See table 244, table 245	DEFAULT = POD12D	Input IO standard

Table 244 GTP_OUTBUFTCO_E1 list of valid values for the parameter (Titan2 HPIO)

GTP_OUTBUFTCO_E1
IOSTANDARD
SSTL18D_I
SSTL18D_II
SSTL15D_I
SSTL15D_II
SSTL135D_I
SSTL135D_II
SSTL12D
HSTL12D_I
HSTL15D_I
HSTL15D_II
HSTL18D_I
HSTL18D_II
HSUL12D
POD12D
SSTL18D_I_CAL
SSTL18D_II_CAL
SSTL15D_I_CAL
SSTL15D_II_CAL
SSTL135D_I_CAL
SSTL135D_II_CAL
SSTL12D_CAL
HSTL12D_I_CAL
HSTL15D_I_CAL
HSTL15D_II_CAL
HSTL18D_I_CAL
HSTL18D_II_CAL
HSUL12D_CAL
POD12D_CAL

Table 245 GTP_OUTBUFTCO_E1 list of valid values for the parameter (Titan2 HRIO)

GTP_OUTBUFTCO_E1
IOSTANDARD
SSTL18D_I
SSTL18D_II
SSTL15D_I
SSTL15D_II
SSTL135D_I

GTP_OUTBUFTCO_E1
IOSTANDARD
SSTL135D_II
HSTL15D_I
HSTL15D_II
HSTL18D_I
HSTL18D_II
HSUL12D

5. Instantiation template

```
GTP_OUTBUFTCO_E1 #(
.IOSTANDARD("DEFAULT")
) GTP_OUTBUFTCO_E1_inst (
.O(), // OUTPUT
.OB(),// OUTPUT
.I(), // INPUT
.IB(),// INPUT
.T(), // INPUT
.TB() // INPUT
);
```

(24) GTP_ISERDES_E2 instruction

1. Supported devices

Table 246 GTP_ISERDES_E2 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

The IOL flexibly supports various application interfaces. In addition to the commonly used direct input/output and input/output registers, the IOL has the function of data input and output rate conversion. GTP_ISERDES_E2 is used for data processing in the input direction, supports direct input and output, ILATCH, IDFF, networking SDR/DDR, oversample, Low-Speed MEMORY DDR, High-Speed MEMORY DDR, and supports SDR1TO2, SDR1TO3, SDR1TO4, SDR1TO5, SDR1TO6, SDR1TO7 , SDR1TO8, DDR1TO2_SAME_PIPELINED, DDR1TO2_SAME_EDGE, DDR1TO2_OPPOSITE_EDGE, DDR1TO4, DDR1TO6, DDR1TO8, DDR1TO10, DDR1TO14, HMDDR1TO4, HMDDR1TO8, LMDDR1TO4, LMDDR1TO8 and OVERSAMPLE mode rate conversion. For detailed instructions, please refer to "UG050006_Titan2 Series FPGA Input and Output Interface (IO) User Guide". Its structural block diagram is as follows:

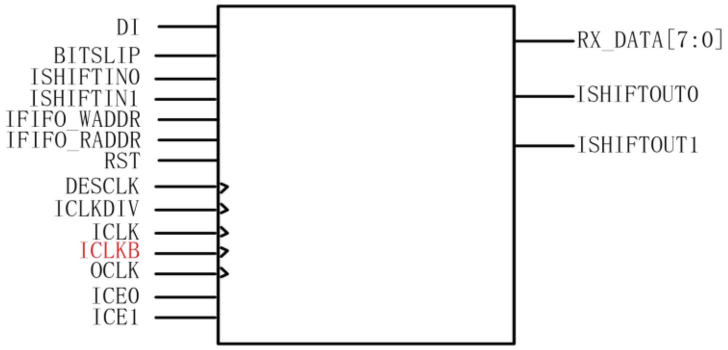


Figure 86 GTP_ISERDES_E2 structure diagram

3. Port description

Table 247 GTP_ISERDES_E2 port description

Signal Port	Input / Output	Describe
DI	Input	Data input
BITSLIP	Input	Data input
ISHIFTIN0	Input	Cascade input signal
ISHIFTIN1	Input	Cascade input signal
IFIFO_WADDR	Input	FIFO write address DQS falling edge trigger (Gray code)
IFIFO_RADDR	Input	FIFO read address (Gray code)
RST	Input	ILOGIC's local reset signal
DESCLK	Input	ILOGIC Deserialization High Speed Clock
ICLKDIV	Input	LOGIC low speed clock
ICLK	Input	ILOGIC first level high speed clock
ICLKB	Input	ILOGIC first-level high-speed clock reverse input (only as an emulation port, software does not map)
OCLK	Input	OLOGIC output stage high speed clock
ICE0	Input	ILOGIC clock enable signal
ICE1	Input	ILOGIC clock enable signal
DO	Output	Output signal after deserialization
ISHIFTOUT0	Output	Output signal after deserialization
ISHIFTOUT1	Output	Output signal after deserialization

4. Parameter description

Table 248 GTP_ISERDES_E2 parameter list description

	Parameter Type	Settings	Defaults	Functional description
HPIO	String	"TRUE" "FALSE"	"FALSE"	"TRUE" maps to HPIO, "FALSE" maps to HRIO
ISERDES_MODE	String	"SDR1TO2" "SDR1TO3" "SDR1TO4" "SDR1TO5" "SDR1TO6" "SDR1TO7"	SDR1TO4	""SDR1TO2": networking SDR 1:2 deserialization mode ""SDR1TO3": networking SDR 1:3 deserialization mode "SDR1TO4": networking SDR 1:4 deserialization model

	Parameter Type	Parameter Type	Defaults	Functional description
		"SDR1TO8" "ILATCH" "IDFF" "DDR1TO2_SAME_PIPELINED" "DDR1TO2_SAME_EDGE" "DDR1TO2_ OPPO SITE_EDGE" "DDR1TO4" "DDR1TO6" "DDR1TO8" "DDR1TO10" "DDR1TO14" "HMDDR1TO4" "HMDDR1TO8" "LMDDR1TO4" "LMDDR1TO8" "OVERSAMPLE"		"SDR1TO5": networking SDR 1:5 deserialization mode "SDR1TO6": networking SDR 1:6 deserialization mode "SDR1TO7": networking SDR 1:7 deserialization mode "SDR1TO8": networking SDR 1:8 deserialization mode "ILATCH" Latch Input Mode "IDFF" Register Input Mode "DDR1TO2_same_piplelined":networki ng DDR 1:2 same pipelined deserialization mode "DDR1TO2_SAME_EDGE":networkin g DDR 1:2 SAME_EDGE deserialization mode "DDR1TO2_OPPOSITE_EDGE":netw orking DDR 1:2 OPPOSITE_EDGE deserialization mode "DDR1TO4": networking DDR 1:4 deserialization mode "DDR1TO6": networking DDR 1:6 deserialization mode "DDR1TO8": networking DDR 1:8 deserialization mode "DDR1TO10": networking DDR 1:10 deserialization mode "DDR1TO14": networking DDR 1:14 deserialization mode "HMDDR1TO4": High-Speed Memory DDR 1:4 deserialization mode "HMDDR1TO8": High-Speed Memory DDR 1:8 deserialization mode "LMDDR1TO4": Low-Speed Memory DDR 1:4 deserialization mode "LMDDR1TO8": Low-Speed Memory DDR 1:8 deserialization mode "OVERSAMPLE" deserialization mode
CASCADE_MODE	String	"MASTER" "SLAVE"	"MASTER"	Deserialized cascaded master mode
BITSLIP_EN	String	"FALSE" "TRUE"	"FALSE"	bitslip enable
GRS_EN	String	"TRUE" "FALSE"	"TRUE"	Global reset/set enable
NUM_ICE	Constance	1'b0 1'b1	1'b0	Ice number selection
GRS_TYPE_Q0	String	"RESET" "SET"	"RESET"	DFF0 global asynchronous reset/set result in ILOGIC gear;
GRS_TYPE_Q1	String	"RESET" "SET"	"RESET"	DFF1 global asynchronous reset/set result in ILOGIC gear;

	Parameter Type	Settings	Defaults	Functional description
GRS_TYPE_Q2	String	"RESET" "SET"	"RESET"	DFF2 global asynchronous reset/set result in ILOGIC gear;
GRS_TYPE_Q3	String	"RESET" "SET"	"RESET"	DFF3 global asynchronous reset/set result in ILOGIC gear;
LRS_TYPE_Q0	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_RESET"	DFF0 local asynchronous/synchronous/reset/set result in ILOGIC gear
LRS_TYPE_Q1	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_RESET"	DFF1 local asynchronous/synchronous/reset/set result in ILOGIC gear
LRS_TYPE_Q2	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_RESET"	DFF2 local asynchronous/synchronous/reset/set result in ILOGIC gear
LRS_TYPE_Q3	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_RESET"	DFF3 local asynchronous/synchronous/reset/set result in ILOGIC gear

5. Instantiation template

```
GTP_GRS GRS_INST(grs_n);
GTP_ISERDES_E2 # (
.HPIO("FALSE"),
.ISERDES_MODE ("SDR1TO4"),
.CASCADE("MASTER"),
.BITSLIP_EN("FALSE"),
.GRS_EN ("TRUE"),
.NUM_ICE(1'b0),
.GRS_TYPE_Q0("RESET"),
.GRS_TYPE_Q1("RESET"),
.GRS_TYPE_Q2("RESET"),
.GRS_TYPE_Q3("RESET"),
.LRS_TYPE_Q0("ASYNC_RESET"),
.LRS_TYPE_Q1("ASYNC_RESET"),
.LRS_TYPE_Q2("ASYNC_RESET"),
.LRS_TYPE_Q3("ASYNC_RESET")
) gtp_iserdes_inst (
.RST(rst),
.ICE0(ice0),
.ICE1(ice1),
.DESCLK (desclk),
```

```
.ICLK (iclk),
.ICLKB (iclkb),
.OCLK (oclk),
.ICLK DIV(iclkdiv),
.DI (di),
.BITSLIP(bitslip),
.ISHIFTIN0(ishiftin0),
.ISHIFTIN1(ishiftin1),
.IFIFO_WADDR(ififo_waddr),
.IFIFO_RADDR(ififo_raddr),
.DO(DO),
.ISHIFTOUT0(ishiftout0),
.ISHIFTOUT1(ishiftout1)
);
```

(25) GTP_OSERDES_E2 instruction

1. Supported devices

Table 249 GTP_OSERDES_E2 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

IOL flexibly supports various application interfaces. In addition to the commonly used direct input/output and input/output registers, IOL has the function of data input and output rate conversion. The GTP_OSERDES_E2 data output conversion module supports SDR2TO1, SDR3TO1, SDR4TO1, SDR5TO1, SDR6TO1, SDR7TO1, SDR8TO1, DDR2TO1_SAME_EDGE, DDR2TO1_OPPOSITE_EDGE, DDR4TO1, DDR6TO1, DDR8TO1, DDR10TO1, DDR14TO1, HMSDR2TO1, HMSDR4TO1, HMSDR8TO1, OLATCH, ODFF mode rate conversion. For detailed instructions, please refer to "UG050006_Titan2 Series FPGA Input-Output Interface (IO) User Guide". Structure The block diagram is as follows:

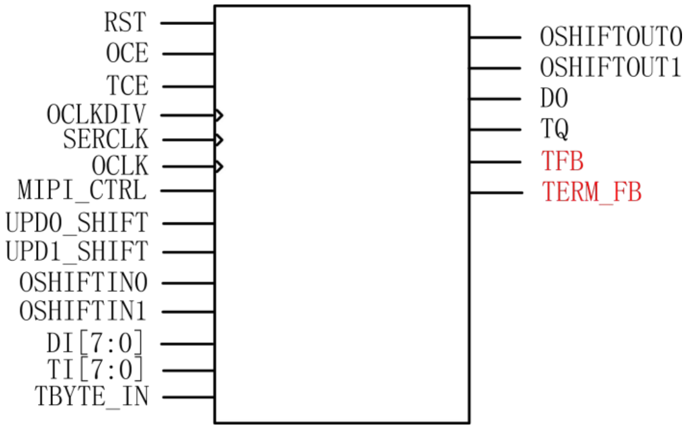


Figure 87 GTP_OSERDES _E2 structure diagram

3. Port description

Table 250 GTP_OSERDES_E2 port description

Port signal	Input / Output	Data width	Describe
RST	Input	1	Local reset signal
OCE	Input	1	Output module clock enable signal
TCE	Input	1	Tri-state module clock enable signal
OCLKDIV	Input	1	OLOGIC low speed clock
SERCLK	Input	1	OLOGIC serialized clock
OCLK	Input	1	OLOGIC output stage high speed clock
MIPI_CTRL	Input	1	Output mipi (low speed/high speed) control signal
UPD0_SHIFT	Input	1	OLOGIC's UPD0 position moves
UPD1_SHIFT	Input	1	OLOGIC's UPD1 position moves
OSHIFTIN0	Input	1	Cascade input signal
OSHIFTIN1	Input	1	Cascade input signal
DI	Input	8	Parallel input data signal
TI	Input	2	Parallel input tri-state control signal
TBYTE_IN	Input	1	Can control a byte tri-state control input signal
OSHIFTOUT0	Output	1	Cascade input signal
OSHIFTOUT1	Output	1	Cascade input signal
DO	Output	1	Data output
TQ	Output	1	Three-state control output (to IOB)
TFB	Output	1	Three-state control output (to internal logic)
TERM_FB	Output	1	Terminal control output (to internal logic)

4. Parameter description

Table 251 GTP_OSERDES_E2 parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
HPIO	String	"TRUE", "FALSE"	"FALSE"	TRUE means OSERDES using HPIO, otherwise HRIO
ODELAY_EN	String	"TRUE", "FALSE"	"FALSE"	TRUE indicates that the output MUX data source uses ODELAY
TDELAY_EN	String	"TRUE", "FALSE"	"FALSE"	TRUE indicates that the output MUX data source uses TDELAY
OSERDES_MODE	String	"SDR2TO1" "SDR3TO1" "SDR4TO1" "SDR5TO1" "SDR6TO1" "SDR7TO1" "SDR8TO1" "OLATCH" "ODFF"	SDR4TO1	"SDR2TO1": Generic SDR 2:1 serialization mode "SDR3TO1": Generic SDR 3:1 serialization mode "SDR4TO1": Generic SDR 4:1 serialization mode "SDR5TO1": Generic SDR 5:1 serialization mode "SDR6TO1" Generic SDR 6:1 serialization mode "SDR7TO1": Generic SDR 7:1 serialization mode "SDR8TO1": Generic SDR 8:1 serialization mode "OLATCH": Latch output mode "ODFF" Register Output Mode

Parameter name	Parameter Type	Settings	Defaults	Functional description
		"DDR2TO1_SAME_ EDGE" "DDR2TO1_OPPOSITE_EDGE" "DDR4TO1" "DDR6TO1" "DDR8TO1" "DDR10TO1" "DDR14TO1" "HMSDR4TO1" "HMSDR8TO1"		"DDR2TO1_SAME_EDGE": Generic DDR 2:1 serialization mode "DDR2TO1_OPPOSITE_EDGE": Generic DDR 2:1 serialization mode "DDR4TO1": Generic DDR 4:1 serialization mode "DDR6TO1": Generic DDR 6:1 serialization mode "DDR8TO1": Generic DDR 8:1 serialization mode "DDR10TO1": Generic DDR 10:1 serialization mode "DDR14TO1": Generic DDR 14:1 serialization mode "HMSDR4TO1": High-Speed Memory DDR 4:1 serialization mode "HMSDR8TO1": High-Speed Memory DDR 8:1 serialization mode
TSERDES_MODE	String	"SDR2TO1" "SDR4TO1" "SDR8TO1" "TLATCH" "TDFF" "DDR2TO1_SAME_ EDGE" "DDR2TO1_OPPOSITE_EDGE" "DDR4TO1" "DDR8TO1" "HMSDR4TO1" "HMSDR8TO1"	SDR4TO1	"SDR2TO1": Generic SDR 2:1 serialization mode "SDR4TO1": Generic SDR 4:1 serialization mode "SDR8TO1": Generic SDR 8:1 serialization mode "TLATCH": Latch output mode "TDFF" register output Mode "DDR2TO1_SAME_EDGE": Generic DDR2:1 serialization mode "DDR2TO1_OPPOSITE_EDGE": GenericDDR 2:1 serialization mode "DDR4TO1": Generic DDR 4:1 serialization mode "DDR8TO1": Generic DDR 8:1 serialization mode "HMSDR4TO1" : High-Speed Memory DDR 4:1 serialization mode "HMSDR8TO1": High-Speed Memory DDR8:1 serialization mode
UPD0_SHIFT_EN	String	"FALSE" "TRUE"	"FALSE"	OLOGIC upd0 position control enable
UPD1_SHIFT_EN	String	"FALSE" "TRUE"	"FALSE"	OLOGIC upd1 position control enable
TUPD0_SHIFT_EN	String	"FALSE" "TRUE"	"FALSE"	TSERDES upd0 position control enable
TUPD1_SHIFT_EN	String	"FALSE" "TRUE"	"FALSE"	TSERDES upd1 position control enable
INIT_SET[1:0]	Constance	2'b00, 2'b01, 2'b10, 2'b11	2'b00	Static control of upd0 and upd1 positions in OSERDES 2'b00: default position; 2'b01: move forward by one cycle; 2'b10: move forward two cycles; 2'b11: move forward three cycles;
TINIT_SET[1:0]	Constance	2'b00, 2'b01, 2'b10, 2'b11	2'b00	Static control of upd0 and upd1 positions in TSERDES 2'b00: default position; 2'b01: move forward by one cycle; 2'b10: move forward two cycles; 2'b11: move forward three cycles;

Parameter name	Parameter Type	Settings	Defaults	Functional description
LRS_TYPE_DQ2	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_RESET"	DFF2 local reset/set junction in OLOGIC data gear;
LRS_TYPE_DQ3	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_ RESET"	DFF3 local reset/set result in OLOGIC data gear;
GRS_TYPE_TQ	String	"RESET" "SET"	"RESET"	Global reset results of DFF0, DFF1, DFF2, DFF3, DFF4 in OLOGIC tri-state gear;
LRS_TYPE_TQ0	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_ RESET"	DFF0 local reset/set result in OLOGIC tri-state gear;
LRS_TYPE_TQ1	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_ RESET"	DFF1 local reset/set result in OLOGIC tri-state gear;
LRS_TYPE_TQ2	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_ RESET"	DFF2 local reset/set result in OLOGIC tri-state gear;
LRS_TYPE_TQ3	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_ RESET"	DFF3 local reset/set result in OLOGIC tri-state gear;*
TSERDES_EN	String	"FALSE" "TRUE"	"FALSE"	TSERDES enable *Note: When oserdes and tserdes are used together, oserdes TSERDES_EN="FALSE"; TSERDES_EN="TRUE" for tserdes; TSERDES_EN="FALSE" when only using oserdes
TRI_EN	String	"FALSE" "TRUE"	"FALSE"	Tri-state control enable IOLHR: *Note: When oserdes and tserdes are used together, oserdes TRI_EN = "TRUE"; tserdes TRI_EN= "FALSE"; When only using oserdes, enable the tri-state function TRI_EN = "TRUE", do not enable the tri-state function TRI_EN = "FALSE". IOLHP: Open when using TSERDES, TRI_EN= "TRUE"; close if not in use, TRI_EN="FALSE";
TBYTE_EN	String	"FALSE" "TRUE"	"FALSE"	BYTE control enable *Note: When oserdes and tserdes are used together, TBYTE_EN="TRUE" of oserdes, and tserdes TBYTE_EN="FALSE"
TERMBYTE_EN	String	"FALSE" "TRUE"	"FALSE"	TERM BYTE control enable

Parameter Name	Parameter Type	Settings	Defaults	Functional description
OCASCADE_EN	String	"FALSE" "TRUE"	"FALSE"	OSERDES extension enable
TERM_EN	String	"FALSE" "TRUE"	"FALSE"	Terminal control signal TERM enable
TERM_OFF_SET	Constance	2'b00, 2'b01, 2'b10, 2'b11	2'b00	TERM signal off (falling edge) bias 2'b00: synchronized with tri-state signal; 2'b01: Advance to TI data update (not supported for now); 2'b10: advance to UPD0 effective (not supported for now); 2'b11: Advance to UPD1 is valid (not supported for now);
TERM_ON_SET	Constance	2'b00, 2'b01, 2'b10, 2'b11	2'b00	TERM signal on (rising edge) bias 2'b00: synchronized with the tri-state signal; 2'b01: delay one cycle; 2'b10: delay two cycles; 2'b11: delay three cycles;
GRS_EN	String	"FALSE" "TRUE"	"TRUE"	Global reset enable
GRS_TYPE_DQ	String	"RESET" "SET"	"RESET"	Global reset results of DFF0, DFF1, DFF2, DFF3, DFF4 in OLOGIC data gear;
LRS_TYPE_DQ0	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_ RESET"	DFF0 local reset/set result in OLOGIC data gear;
LRS_TYPE_DQ1	String	"ASYNC_RESET" "ASYNC_SET" "SYNC_RESET" "SYNC_SET"	"ASYNC_ RESET"	DFF1 local reset/set result in OLOGIC data gear;
MIPI_EN	String	"FALSE" "TRUE"	"FALSE"	MIPI enable

Note: The blue background color parameter only supports Titan22 IOLHP

5. Instantiation template

```
GTP_GRS GRS_INST(grs_n); //全局复位控制信号为隐式 Instance 调用，需同时例化 GRS_INST

GTP_OSERDES_E2 # (
.GRS_EN ("TRUE"),
.HPIO("FALSE"),
.ODELAY_EN("FALSE"),
.TDELAY_EN("FALSE"),
. OSERDES_MODE ("SDR4TO1"),
. TSERDES_MODE ("SDR4TO1"),
. UPD0_SHIFT_EN ("FALSE"),
. UPD1_SHIFT_EN ("FALSE"),
. TUPD0_SHIFT_EN ("FALSE"),
. TUPD1_SHIFT_EN ("FALSE"),
. INIT_SET (2'b00),
. TINIT_SET (2'b00),
```

```
. GRS_TYPE_DQ ("RESET"),
. LRS_TYPE_DQ0 ("ASYNC_RESET"),
. LRS_TYPE_DQ1 ("ASYNC_RESET"),
. LRS_TYPE_DQ2 ("ASYNC_RESET"),
. LRS_TYPE_DQ3 ("ASYNC_RESET"),
. GRS_TYPE_TQ ("RESET"),
. LRS_TYPE_TQ0 ("ASYNC_RESET"),
. LRS_TYPE_TQ1 ("ASYNC_RESET"),
. LRS_TYPE_TQ2 ("ASYNC_RESET"),
. LRS_TYPE_TQ3 ("ASYNC_RESET"),
. TRI_EN ("FALSE"),
. TSERDES_EN ("FALSE"),
. TBYTE_EN ("FALSE"),
. TERMBYTE_EN ("FALSE"),
. MIPI _EN ("FALSE"),
. OCASCADE_EN ("FALSE"),
. TERM_EN("FALSE"),
. TERM_OFF_SET(2'b00),
. TERM_ON_SET(2'b00)
) GTP_OSERDES_E2_INST (
. RST (RST),
. OCE (OCE),
. TCE (TCE),
. OCLKDIV (OCLKDIV),
. SERCLK (SERCLK),
. OCLK (OCLK),
. MIPI_CTRL (MIPI_CTRL),
. UPD0_SHIFT (UPD0_SHIFT),
. UPD1_SHIFT (UPD1_SHIFT),
. OSHIFTIN0 (OSHIFTIN0),
. OSHIFTIN1 (OSHIFTIN1),
. DI (DI),
. TI (TI),
. TBYTE_IN (TBYTE_IN),
. OSHIFTOUT0 (OSHIFTOUT0),
. OSHIFTOUT1 (OSHIFTOUT1),
. DO (DO),
. TQ (TQ),
. TFB(TFB),
. TERM_FB(TERM_FB)
);
```

(26) GTP_IODELAY_E2 instruction

1. Supported devices

Table 252 GTP_IODELAY_E2 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IODELAY_E2 is a data delay unit that supports dynamic or static delay control, including coarse adjustment and fine adjustment, and supports a maximum delay of 247 steps, a total of 248 delay combinations. For detailed instructions, please refer to "UG050006_Titan2 Series FPGA Input and Output Interface (IO) User Guide ". Its structural block diagram is as follows:

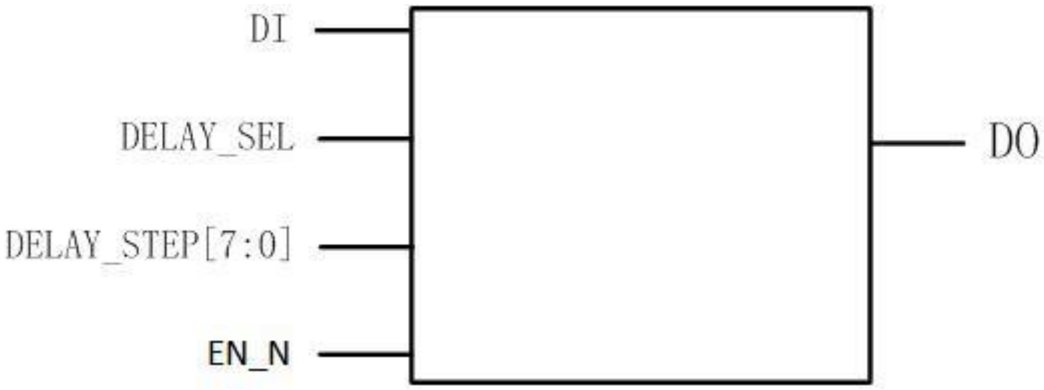


Figure 88 GTP_IODELAY_E2 structure diagram

3. Port description

Table 253 GTP_IODELAY_E2 port description

Port signal	Input / Output	Bit width	Describe
DI	Input	1 bit	Data input
DELAY_SEL	Input	1 bit	Cascade selection
DELAY_STEP	Input	8 bit	Iodelay delay control code configuration (gray code)
DO	Output	1 bit	Data output
EN_N	Input	1 bit	Active low signal

4. Parameter description

Table 254 GTP_IODELAY_E2 parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
DELAY_STEP_SEL	String	"PARAMETER" "PORT"	"PARAMETER"	"PARAMETER": select static configuration; "PORT": dynamic configuration
DELAY_STEP_VALUE	Constant	8'h00~8'hF7	8'h00	Static configuration of iodelay delay control code (Gray code)
TDLY	String	"FALSE" "TRUE"	"FALSE"	"TRUE": Map Tian2 HPIO Tdelay function

5. Instantiation template

```
GTP_IODELAY_E2 # (
.TDLY("FALSE"),
```

```
.DELAY_STEP_SEL ("PARAMETER"),
.DELAY_STEP_VALUE( 8'h00 )
) GTP_IODELAY_E2_inst (
.EN_N (en_n),
.DI (di),
.DELAY_SEL (delay_sel),
.DELAY_STEP (delay_step),
.DO (do)
);
```

(27) GTP_ZEROHOLDDELAY instruction

1. Supported devices

Table 255 GTP_ZEROHOLDDELAY device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_ZEROHOLDDELAY is a data delay unit, supports static delay control, supports a maximum delay of 15 steps, and has a total of 16 delay combinations. For detailed instructions, please refer to "UG050006_Titan2 Series FPGA Input and Output Interface (IO) User Guide". Its structure diagram is as follows :



Figure 89 GTP_ZEROHOLDDELAY structure diagram

3. Port description

Table 256 GTP_ZEROHOLDDELAY port description

Port signal	Input / Output	Describe
DI	Input	Data input
DO	Output	Data output

4. Parameter description

Table 257 GTP_ZEROHOLDDELAY parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description

ZHOLD_SET	String	"NODELAY", "100ps", "200ps", "300ps", "400ps", "500ps","600ps", "700ps", "800ps", "900ps", "1000ps", "1100ps", "1200ps", "1300ps", "1400ps", "1500ps"	"NODELAY"	zeroholddelay delay control code

5. Instantiation template

```
GTP_ZEROHOLDDELAY # (
.ZHOLD_SET ("NODELAY")
) GTP_ZEROHOLDDELAY_inst (
.DI (di),
.DO (do)
);
```

6. Clock-related GTP Instructions

(1) GTP_CLKBUFG instruction

1. Supported devices

Table 258 GTP_CLKBUFG device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_CLKBUFG provides a simple clock BUFFER function, and the global clock can be realized by instantiating GTP_CLKBUFG.

3. Port description

Table 259 GTP_CLKBUFG port description

Port signal	Input / Output	Describe
CLKIN	Input	Input clock
CLKOUT	Output	Output clock

4. Instantiation template

```
GTP_CLKBUFG U_CLKBUFG (
.CLKOUT        (clkout),
.CLKIN         (clkin )
);
```

5. Detailed function introduction

GTP_CLKBUFG provides a simple clock BUFFER function, and the global clock can be realized by instantiating GTP_CLKBUFG.

(2) GTP_CLKBUFGCE Instructions

1. Supported devices

Table 260 GTP_CLKBUFGCE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_CLKBUFGCE belongs to the clock BUFFER with the enable control port, and the global clock can also be realized by instantiating GTP_CLKBUFGCE.

3. Port description

Table 261 GTP_CLKBUFGCE port description

Port signal	Input / Output	Illustrate
CLKIN	Input	Input clock
CE	Input	Clock enable, active high enable
CLKOUT	Output	Output clock

4. Parameter description

Table 262 GTP_CLKBUFGCE parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
DEFAULT_VALUE	character	1'b0, 1'b1	1'b0	When CE=0, output DEFAULT_VALUE; When CE=1, output CLKIN;
SIM_DEVICE	string	"TITAN" "LOGOS" "COMPACT" "LOGOS2" "TITAN2"	"TITAN"	When SIM_DEVICE = "TITAN", the clock output is enabled with glitch. Enable clock output without glitch when SIM_DEVICE = "LOGOS2" or "LOGOS" or "COMPACT" or "TITAN2"

5. Instantiation template

```
GTP_CLKBUFGCE
#(.DEFAULT_VALUE       (1'b0      )  //1'b0; 1'b1
.SIM_DEVICE("TITAN")
) I_GTP_CLKBUFGCE  (
.CLKIN        (clk      ),
.CE           (ce       ),
.CLKOUT       (clkout )
);
```

6. Detailed function introduction

When DEFAULT_VALUE=1'b0, the timing diagram is as follows:

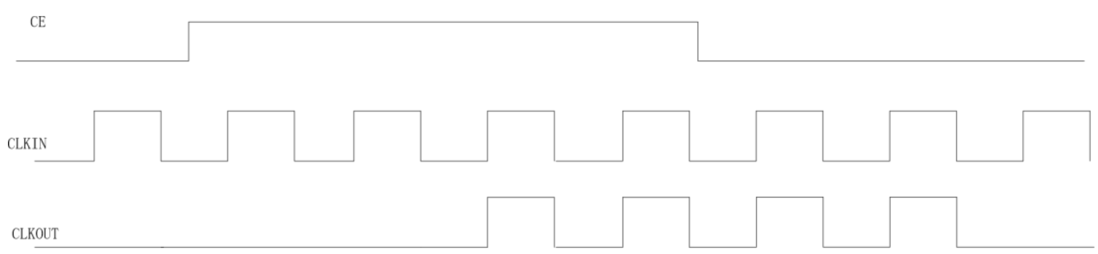


Figure 90 GTP_CLKBUFGCE timing diagram (DEFAULT_VALUE=0)

When CE=1, the output clock is equal to the input clock. When CE jumps and switches from high level to low level, after detecting the falling edge of CLKIN twice, CLKOUT outputs low level. Until the next time CE switches back to high level, then it needs to detect two more falling edges of CLKIN, and the CLKOUT output switches from low level to CLKIN.

When DEFAULT_VALUE=1'b1, the timing diagram is as follows:

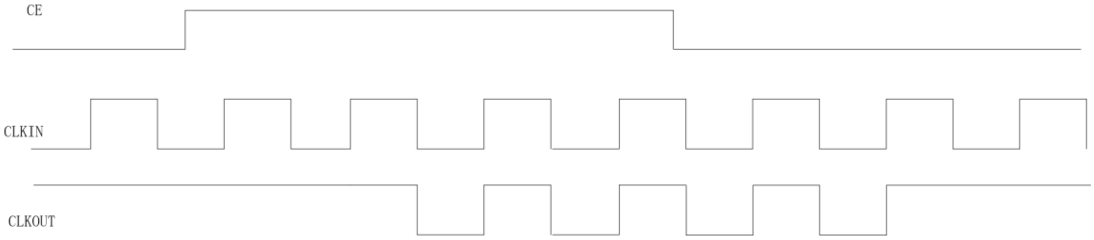


Figure 91 GTP_CLKBUFGCE timing diagram (DEFAULT_VALUE=1)

When CE=1, the output clock is equal to the input clock. When CE jumps and switches from high level to low level, it needs to detect the rising edge of CLKIN twice, and CLKOUT outputs high level. Until the next time CE switches back to high Level, and then need to detect two rising edges of CLKIN, CLKOUT output switches from high level to CLKIN.

(3) GTP_CLKBUFGMUX instruction

1. Supported devices

Table 263 GTP_CLKBUFGMUX device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_CLKBUFGMUX can realize dynamic switching between two global clock input sources, users can choose forced switching (corresponding to TRIGGER_MODE="NORMAL"), or deburring switching triggered by the falling edge of the clock (corresponding to TRIGGER_MODE="NEGEDGE") It is worth noting that forced switching may introduce glitches, and the user needs to reset the relevant logic after switching to achieve the expected function.

3. Port description

Table 264 GTP_CLKBUFGMUX port description

Port signal	Input / Output	Illustrate
CLKIN0	Input	Input clockCLKIN0
CLKIN1	Input	Input clockCLKIN1
SEL	Input	Clock selection signal, 0 selects CLKIN0; 1 selects CLKIN1;
CLKOUT	Output	Output clock

4. Parameter description

Table 265 GTP_CLKBUFGMUX parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
TRIGGER_MODE	<string>	"NORMAL" "NEGEDGE" "POSEDGE"	"NORMAL"	(1) "NORMAL": In this mode, freely switch the input clock (CLKIN0 and CLKIN1), without providing GLITCHLESS function (2) "NEGEDGE": In this mode, freely switch the input clock (CLKIN0 and CLKIN1), provide GLITCHLESS function, Triggered by the falling edge of the clock. (3) "POSEDGE": In this mode, freely switch the input clock (CLKIN0 and CLKIN1), provide GLITCHLESS function, triggered by the rising edge of the clock
SIM_DEVICE	string	"TITAN" "LOGOS" "COMPACT" "LOGOS2" "TITAN2"	"TITAN"	Simulation model device identification "TITAN": when triggered by a rising edge or a falling edge, switch between two clock domains with one beat respectively "LOGOS", "COMPACT", "LOGOS2" and "TITAN2": when triggered by a rising or falling edge, two beats in the two clock domains before switching

Note: 1.Titan2 series support "NORMAL", "NEGEDGE", "POSEDGE" three modes, other devices support please refer to the corresponding clock resource user guide.

2. When the value of TRIGGER_MODE is "NEGEDGE" or "POSEDGE", it is necessary to collect the rising or falling edge of the clock in the two clock domains to switch the switching clock, so both ports of CLKIN0/CLKIN1 must have the rising or falling edge of the clock Only when the GTP can switch the clock normally.

5. Instantiation template

```
GTP_CLKBUFGMUX #(
.TRIGGER_MODE ("NORMAL")
.SIM_DEVICE("TITAN")
)I_CLKBUFGMUX(
.CLKIN0       (clkin0   ),
.CLKIN1       (clkin1   ),
.SEL          (sel      ),
.CLKOUT       (clkout   )
);
```

6. Detailed function introduction

Instantiating GTP_CLKBUFGMUX can realize dynamic switching between two global clock input sources. Users can choose forced switching (corresponding to TRIGGER_MODE="NORMAL") or deburring switching triggered by the falling edge of the clock (corresponding to TRIGGER_MODE=" NEGEDGE "). It is worth noting that forced switching may introduce glitches, and the user needs to reset the relevant logic after switching to achieve the expected function. When the value of the parameter TRIGGER_MODE is "NORMAL", CLKOUT is switched immediately after the SEL signal jumps, and the corresponding timing diagram is as follows:

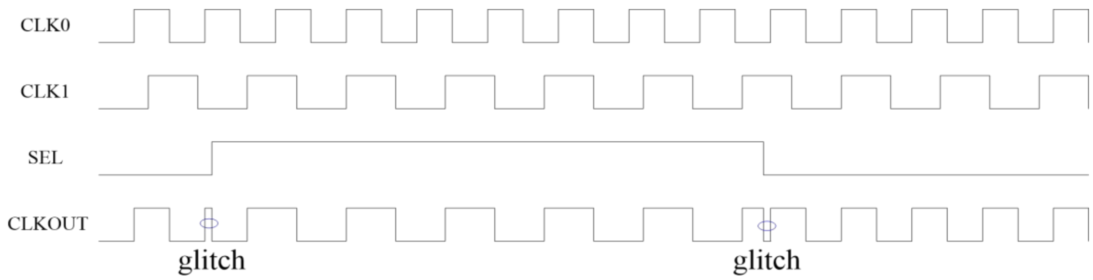


Figure 92 GTP_CLKBUFGMUX. Timing diagram of the parameter TRIGGER_MODE="NORMAL"

When the value of the parameter TRIGGER_MODE is "NEGEDGE" and the value of the parameter SIM_DEVICE is "TITAN2", SEL changes from 0 to 1, and it needs to detect the falling edge of the current clock CLKIN0 twice (during the process of detecting the rising edge of the current clock, CLKOUT output maintains the current clock), and then detect the falling edge of the switched clock CLKIN1 twice (during the process of detecting the falling edge of the switched clock, CLKOUT outputs low level), after the detection is completed, CLKOUT is switched from CLKIN0 to CLKIN1; otherwise, when SEL When changing from 1 to 0, first detect the falling edge of CLKIN1 twice (during the process of detecting the rising edge of the current clock, the CLKOUT output maintains the current clock), and then detect the falling edge of CLKIN0 twice (detect the falling edge of the switched clock CLKIN0 During the process, CLKOUT outputs low level), and after the detection is completed, CLKOUT is switched from CLKIN1 to CLKIN0. The corresponding sequence diagram is as follows:

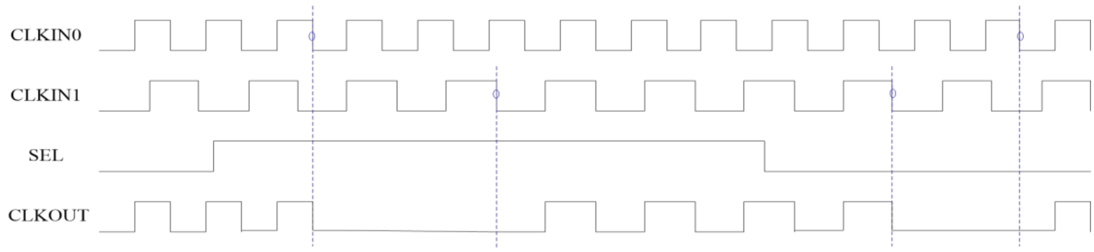


Figure 93 GTP_CLKBUFGMUX. Timing diagram of the parameter TRIGGER_MODE="NEGEDGE"

When the value of the parameter TRIGGER_MODE is "POSEDGE" and SEL changes from 0 to 1, it is necessary to detect the rising edge of the current clock CLKIN0 twice (during the process of detecting the rising edge of the current clock, the output of CLKOUT maintains the current clock), and then detect the Switch the rising edge of clock CLKIN1 twice (during the process of detecting the rising edge of the switched clock, CLKOUT outputs high level), after the detection is completed, CLKOUT is switched from CLKIN0 to CLKIN1; otherwise, when SEL changes from 1 to 0, first detect The rising edge of CLKIN1 is detected twice (during the process of detecting the rising edge of the current clock, CLKOUT output maintains the current clock), and then the rising edge of CLKIN0 is detected twice (during the process of detecting the rising edge of the switched clock CLKIN0, CLKOUT outputs high level) , CLKOUT is switched from CLKIN1 to CLKIN0 after the detection is completed. The corresponding sequence diagram is as follows:

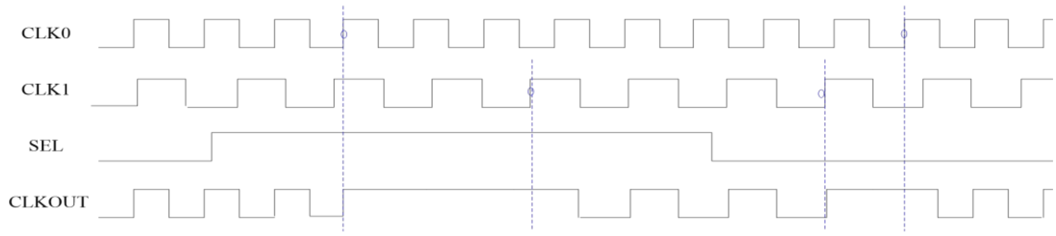


Figure 94 GTP_CLKBUFGMUX. Timing diagram of the parameter TRIGGER_MODE="POSEDGE"

(4) GTP_CLKBUFR instruction

1. Supported devices

Table 266 GTP_CLKBUFR device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_CLKBUFR is a BUFFER of a regional clock, and the regional clock drives the unit logic of a certain region.

3. Port description

Table 267 GTP_CLKBUFR port description

Port signal	Input / Output	Describe
CLKIN	Input	Input clock
CLKOUT	Output	Output clock

4. Instantiation template

```
GTP_CLKBUFR  U_BUFR(
.CLKOUT       (clkout),
.CLKIN        (clkin )
);
```

5. Detailed function introduction

GTP_CLKBUFR is a BUFFER of a regional clock, and the regional clock drives the unit logic of a certain region. The following figure shows the input and output timing diagram of GTP_CLKBUFR

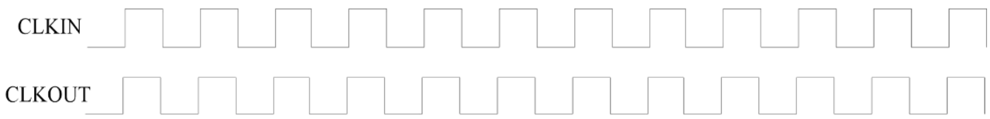


Figure 95 GTP_CLKBUFR timing diagram

(5) GTP_IOCLKBUF instruction

1. Supported devices

Table 268 GTP_IOCLKBUF device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IOCLKBUF is equivalent to the gate function, instantiate GTP_IOCLKBUF to turn off or turn on the IO clock.

3. Port description

Table 269 GTP_IOCLKBUF port description

Port signal	Input / Output	Describe
CLKIN	Input	Input clock
DI	Input	Enable signal
CLKOUT	Output	Output clock

4. Parameter description

Table 270 GTP_IOCLKBUF parameter description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GATE_EN	<string>	<" FALSE ", " TRUE ">	<"FALSE">	Control whether register functions need to be implemented

Note: Titan2 series GATE_EN parameter only supports "FALSE"

5. Instantiation template

```
GTP_IOCLKBUF #(
.GATE_EN      ("FALSE" ) //FALSE; TRUE
) u_IOCLKBUF (
.CLKOUT       (clkout  ),
.CLKIN        (clin    ),
.DI           (di      )
);
```

6. Detailed function introduction

GTP_IOCLKBUF is equivalent to the gate function, instantiate GTP_IOCLKBUF to turn off or turn on the IO clock. When the parameter GATE_EN is "TRUE" or "FALSE", the corresponding timing diagram is as follows.

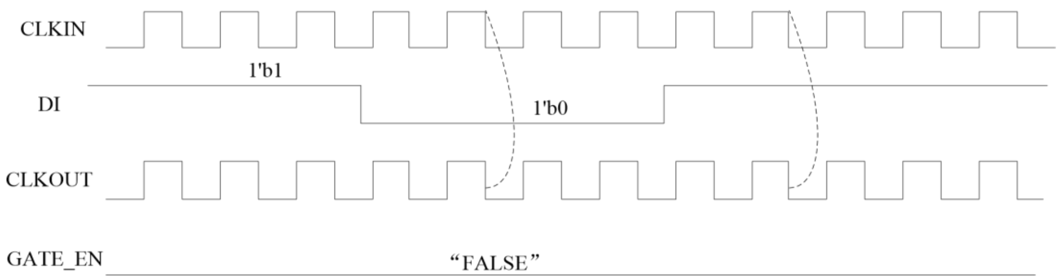


Figure 96 GTP_IOCLKBUF timing diagram of GATEN="FALSE"

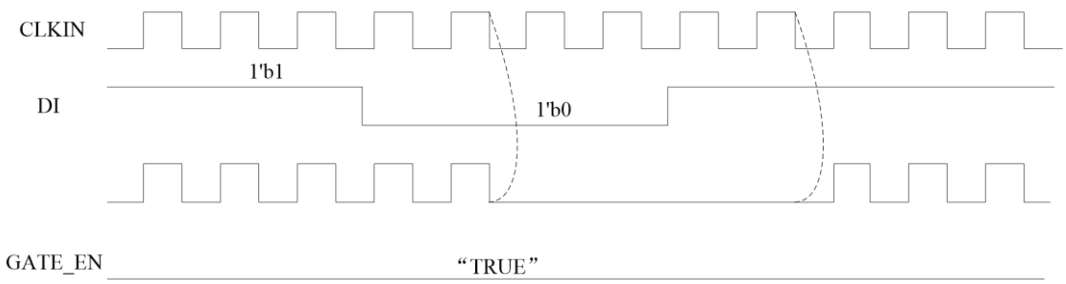


Figure 97 GTP_IOCLKBUF timing diagram for GATEN="TRUE"

(6) GTP_IOCLKDIV_E2 instruction

1. Supported devices

Table 271 GTP_IOCLKDIV_E2 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IOCLKDIV_E2 has 3 inputs CLKIN, RST_N, CE and 1 output CLKDIVOUT and 1 parameter DIV_FACTOR. By controlling DIV_FACTOR, 1-frequency division, 2-frequency division, 3-frequency division, 4-frequency division, 5-frequency division, 6-frequency division, 7-frequency division, 8-frequency division and direct mode (BYPASS) can be realized. In Odd Divide Mode, the high pulse width is one less input clock period than the low pulse width.

CE is asynchronous enable control, when CE = 0, CLKDIVOUT maintains the current level; when CE = 1, CLKDIVOUT outputs effective frequency division;

RST_N is a reset signal that supports asynchronous reset and synchronous release. When RST_N = 0, clear the output clock;

BYPASS mode is not affected by CE or RST_N.

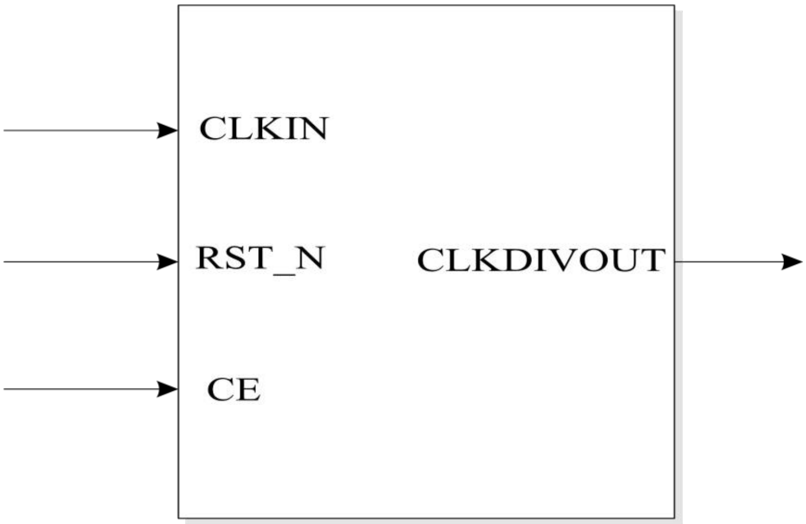


Figure 98 GTP_IOCLKDIV_E2 structure diagram

3. Port description

Table 272 GTP_IOCLKDIV_E2 port description

Port signal	Input / Output	Describe
CLKIN	Input	Input clock
RST_N	Input	Reset signal, active low
CE	Input	Enable control: CE = 1, output clock signal after frequency division CE = 0, the output clock maintains the current level
CLKDIVOUT	Output	Output clock

4. Parameter description

Table 273 GTP_IOCLKDIV_E2 parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
DIV_FACTOR	<string>	"BYPASS" "1" "2" "3" "4" "5""6" "7" "8"	"BYPASS"	Mode selection, including direct mode and 1~8 frequency division mode

5. Instantiation template

```
GTP_IOCLKDIV_E2 #(
.DIV_FACTOR("BYPASS")
) GTP_IOCLKDIV_E2_inst (
.CLKDIVOUT    (clkdivout),
.CE           (ce        ),
.CLKIN        (clkin     ),
.RST_N        (rst_n     )
);
```

(7) GTP_CLKBUFMCE instruction

1. Supported devices

Table 274 GTP_CLKBUFMCE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_CLKBUFMCE is a cross-clock region (vertical direction) clock buffer with an enable port, including 2 input ports CLKIN and CE, 1 output port CLKOUT, and 3 configuration parameters CE_TYPE, TRIGGER_MODE and CE_INV.

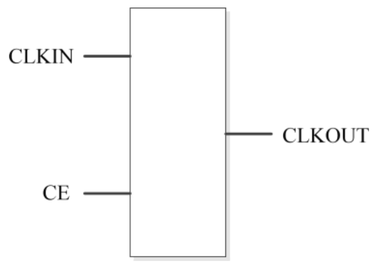


Figure 99 GTP_CLKBUFMCE structure diagram

3. Port description

Table 275 GTP_CLKBUFMCE port description

Port signal	Input / Output	Describe
CLKIN	Input	Input clock signal
CE	Input	Clock enable terminal
CLKOUT	Output	Output clock signal

4. Parameter description

Table 276 GTP_CLKBUFMCE parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
CE_TYPE	<string>	"SYNC", "ASYNC"	"SYNC"	CE_TYPE= "SYNC", support sync enable. CE_TYPE="ASYNC", supports asynchronous enable.
TRIGGER_MODE	<string>	"POSEDGE", "NEGEDGE"	"POSEDGE"	TRIGGER_MODE = "POSEDGE", support clock output initial value equal to 1'b1, rising edge trigger. TRIGGER_MODE = "NEGEDGE", support clock output initial value equal to 1'b0, falling edge trigger.
CE_INV	<string>	"TRUE", "FALSE"	"FALSE"	EN_INV="FALSE", the default CE high level is active. EN_INV="TRUE", the default CE low level is active.

5. Instantiation template

```
GTP_CLKBUFMCE #(
.CE_TYPE("SYNC"),
.CE_INV("FALSE"),
.TRIGGER_MODE("POSEDGE")
) GTP_CLKBUFMCE_inst (
.CLKOUT       (clkout   ),
.CE           (ce       ),
.CLKIN        (clkin    )
);
```

(8) GTP_CLKBUFM instruction

1. Supported devices

Table 277 GTP_CLKBUFM device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_CLKBUFM is a clock buffer across the clock region (vertical direction), with an input CLKIN and an output CLKOUT, without any parameters.

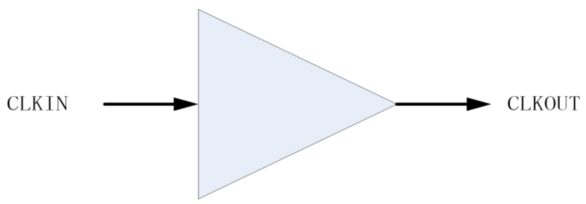


Figure 100 GTP_CLKBUFM structure diagram

3. Port description

Table 278 GTP_CLKBUFM port description

Port signal	Input / Output	Describe

CLKIN	Input	Input clock
CLKOUT	Output	Output clock

4. Instantiation template

```
GTP_CLKBUFM GTP_CLKBUFM_inst (
.CLKOUT(clkout),
.CLKIN(clkin)
);
```

(9) GTP_CLKBUFXCE instruction

1. Supported devices

Table 279 GTP_CLKBUFXCE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_CLKBUFXCE is a cross-clock region (horizontal direction) clock buffer with an enable port, including 2 input ports CLKIN and CE, 1 output port CLKOUT, and 3 configuration parameters CE_TYPE, TRIGGER_MODE and CE_INV.

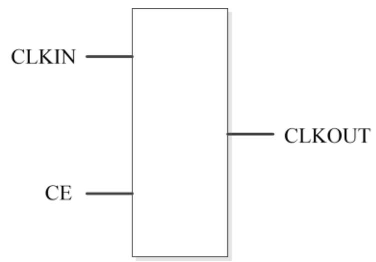


Figure 101 GTP_CLKBUFXCE structure diagram

3. Port description

Table 280 GTP_CLKBUFXCE port description

Port signal	Input / Output	Describe
CLKIN	Input	Input clock signal
CE	Input	Clock enable terminal
CLKOUT	Output	Output clock signal

4. Parameter description

Table 281 GTP_CLKBUFXCE parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
CE_TYPE	<string>	"SYNC", "ASYNC"	"SYNC"	CE_TYPE= "SYNC", support sync enable. CE_TYPE="ASYNC", supports asynchronous enable.
TRIGGER_MODE	<string>	"POSEDGE", "NEGEDGE"	"POSEDGE"	TRIGGER_MODE = "POSEDGE", support clock output initial value equal to 1'b1, rising edge trigger.

				TRIGGER_MODE = "NEGEDGE", support clock output initial value equal to 1'b0, falling edge trigger.
CE_INV	<string>	"TRUE", "FALSE"	"FALSE"	EN_INV= "FALSE", the default CE high level is active. EN_INV="TRUE", the default CE low level is active.

5. Instantiation template

```
GTP_CLKBUFXCE #(
.CE_TYPE("SYNC"),
.CE_INV("FALSE"),
.TRIGGER_MODE("POSEDGE")
) GTP_CLKBUFXCE_inst (
.CLKOUT       (clkout   ),
.CE           (ce       ),
.CLKIN        (clkin    )
);
```

(10) GTP_CLKBUFX instruction

1. Supported devices

Table 282 GTP_CLKBUFX device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_CLKBUFX is a clock buffer across the clock region (horizontal direction), with an input CLKIN and an output CLKOUT, without any parameters.

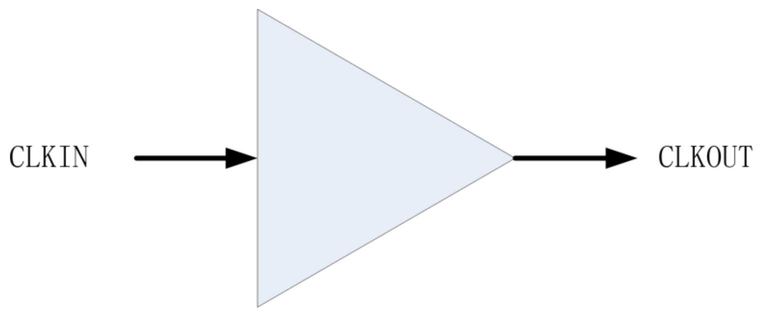


Figure 102 GTP_CLKBUFX structure diagram

3. Port description

Table 283 GTP_CLKBUFX port description

Port signal	Input / Output	Describe
CLKIN	Input	Input clock
CLKOUT	Output	Output clock

4. Instantiation template

```
GTP_CLKBUFX GTP_CLKBUFX_inst (
.CLKOUT(clkout),
.CLKIN(clkin)
);
```

(11) GTP_GPLL instruction

1. Supported devices

Table 284 GTP_GPLL device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

The block diagram of the GPLL structure is shown below.

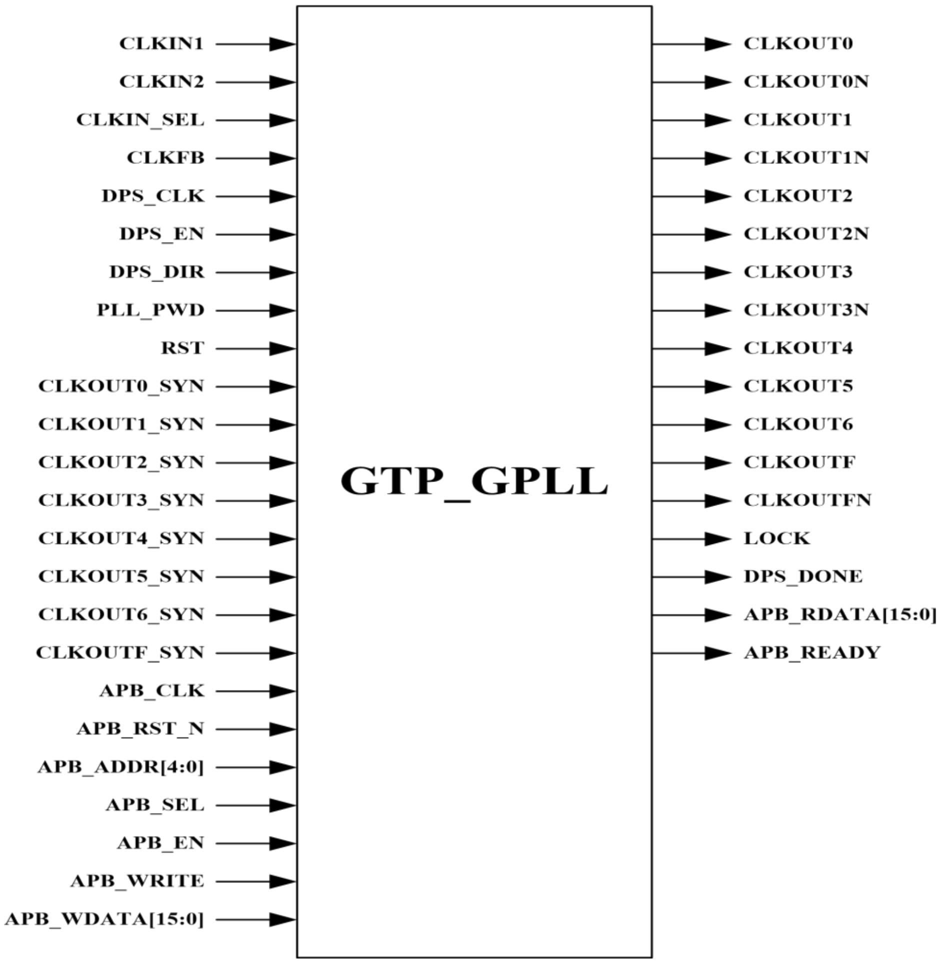


Figure 103 GTP_GPLL structure diagram

3. Port description

Table 285 GTP_GPLL port description

Port signal	Input / Output	Describe

Port signal	Input / Output	Describe
CLKOUT0	Output	PLL No. 0 positive phase output clock;
CLKOUT0N	Output	Inverted output clock of the 0th channel of PLL;
CLKOUT1	Output	PLL No. 1 positive phase output clock;
CLKOUT1N	Output	The first PLL inverted output clock;
CLKOUT2	Output	PLL No. 2 positive phase output clock;
CLKOUT2N	Output	PLL 2nd channel inverting output clock;
CLKOUT3	Output	PLL No. 3 positive phase output clock;
CLKOUT3N	Output	PLL 3rd road inverting output clock;
CLKOUT4	Output	PLL No. 4 output clock;
CLKOUT5	Output	PLL 5th output clock;
CLKOUT6	Output	PLL No. 6 output clock;
CLKOUTF	Output	PLL feedback positive phase output clock;
CLKOUTFN	Output	PLL feedback inverting output clock;
LOCK	Output	The PLL frequency lock indication signal is an asynchronous signal; When the signal is pulled high, it indicates that the PLL feedback clock signal has been locked to the input clock signal;
DPS_DONE	Output	Dynamic interpolation phase shift adjustment indication signal;
APB_RDATA[15:0]	Output	PLL APB interface data bus data output;
APB_READY	Output	PLL APB interface data bus handshake signal; indicates the end of a normal bus cycle;
CLKIN1	Input	PLL reference input clock 1;
CLKIN2	Input	PLL reference input clock 2;
CLKFB	Input	PLL feedback clock;
CLKIN_SEL	Input	Input clock selection signal;
DPS_CLK	Input	Dynamic interpolation phase shift adjustment clock;
DPS_EN	Input	Dynamic interpolation phase shift adjustment enable; active high;
DPS_DIR	Input	Select the direction of dynamic interpolation phase shift adjustment; 1'b0: lagging, 1'b1: leading;
PLL_PWD	Input	PLL Power Down; active high;
RST	Input	PLL reset signal, active high;
CLKOUT0_SYN	Input	CLKOUT0/N output clock enable control; high level active; high level off, low level enable;
CLKOUT1_SYN	Input	CLKOUT1/N output clock enable control; high level active; high level off, low level enable;
CLKOUT2_SYN	Input	CLKOUT2/N output clock enable control; high level active; high level off, low level enable;
CLKOUT3_SYN	Input	CLKOUT3/N output clock enable control; high level active; high level off, low level enable;
CLKOUT4_SYN	Input	CLKOUT4 output clock enable control; high level active; high level off, low level enable
CLKOUT5_SYN	Input	CLKOUT5 output clock enable control; high level active; high level off, low level enabl;
CLKOUT6_SYN	Input	CLKOUT6 output clock enable control; high level active; high level off, low level enable;
CLKOUTF_SYN	Input	CLKOUTF/N output clock enable control; high level active; high level off, low level enable;
APB_CLK	Input	PLL APB interface data bus clock;
APB_RST_N	Input	PLL APB interface data bus asynchronous reset signal, active low;
APB_ADDR	Input	PLL APB interface data bus address;
APB_SEL	Input	PLL APB interface data bus selection signal, select slave device;
APB_EN	Input	PLL APB interface data bus enable signal; indicates the second and subsequent cycles of the transfer;
APB_WRITE	Input	PLL APB interface data bus write enable signal; 1'b0: read operation, 1'b1: write operation;

Port signal	Input / Output	Describe
APB_WDATA[15:0]	Input	PLL APB interface data bus data input;

4. Parameter description

Table 286 GTP_GPLL parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
CLKIN_FREQ	real	10~800	50.0	Input clock frequency configuration,, MHz;
LOCK_MODE	binary	1'b0, 1'b1	1'b0	PLL frequency detection mode configuration; 1'b0: Real-time monitoring of PLL working status; 1'b1: LOCK is kept after PLL is locked, unless reset bit or Power Down;
STATIC_RATIOI	interger	1-80	1	Enter divider ratio static configuration;
STATIC_RATIOM	interger	1-128	1	Feedback M divider ratio static configuration;
STATIC_RATIO0	real	1-128 or 2.000~128.000, step 0.125	1.0	Output divider0 ratio static configuration; Integer frequency division mode, support 1~128; Fractional frequency division mode, support 2.000~128.000;
STATIC_RATIO1	interger	1-128	1	Output divider1 ratio static configuration;
STATIC_RATIO2	interger	1-128	1	Output divider2 ratio static configuration;
STATIC_RATIO3	interger	1-128	1	Output divider3 ratio static configuration;
STATIC_RATIO4	interger	1-128	1	Output divider4 ratio static configuration;
STATIC_RATIO5	interger	1-128	1	Output divider5 ratio static configuration;
STATIC_RATIO6	interger	1-128	1	Output divider6 ratio static configuration;
STATIC_RATIOF	real	1-128 or 2.000~128.000 step 0.125	1.0	Feedback F divider ratio static configuration; Integer frequency division mode, support 1~128; Fractional frequency division mode, support 2.000~128.000;
STATIC_DUTY0	interger	2~255	2	Output divider0 duty static configuration;
STATIC_DUTY1	interger	2~255	2	Output divider1 duty static configuration;
STATIC_DUTY2	interger	2~255	2	Output divider2 duty static configuration;
STATIC_DUTY3	interger	2~255	2	Output divider3 duty static configuration;
STATIC_DUTY4	interger	2~255	2	Output divider4 duty static configuration;
STATIC_DUTY5	interger	2~255	2	Output divider5 duty static configuration;
STATIC_DUTY6	interger	2~255	2	Output divider6 duty static configuration;
STATIC_DUTYF	interger	2~255	2	Feedback F divider duty static configuration;
STATIC_PHASE	interger	0-63	0	Interpolation phase shift phase static configuration;
STATIC_PHASE0	interger	0-7	0	Output divider0 fine phase static configuration;
STATIC_PHASE1	interger	0-7	0	Output divider1 fine phase static configuration;
STATIC_PHASE2	interger	0-7	0	Output divider2 fine phase static configuration;
STATIC_PHASE3	interger	0-7	0	Output divider3 fine phase static configuration;
STATIC_PHASE4	interger	0-7	0	Output divider4 fine phase static configuration;
STATIC_PHASE5	interger	0-7	0	Output divider5 fine phase static configuration;
STATIC_PHASE6	interger	0-7	0	Output divider6 fine phase static configuration;
STATIC_PHASEF	interger	0-7	0	Feedback F divider fine phase static configuration;
STATIC_CPHASE0	interger	0-127	0	Output divider0 coarse phase static configuration;
STATIC_CPHASE1	interger	0-127	0	Output divider1 coarse phase static configuration;

Parameter Name	Parameter Type	Settings	Defaults	Functional description
STATIC_CPHASE2	interger	0-127	0	Output divider2 coarse phase static configuration;
STATIC_CPHASE3	interger	0-127	0	Output divider3 coarse phase static configuration;
STATIC_CPHASE4	interger	0-127	0	Output divider4 coarse phase static configuration;
STATIC_CPHASE5	interger	0-127	0	Output divider5 coarse phase static configuration;
STATIC_CPHASE6	interger	0-127	0	Output divider6 coarse phase static configuration;
STATIC_CPHASEF	interger	0-127	0	Feedback F divider coarse phase static configuration;
CLK_DPS0_EN	string	"FALSE", "TRUE"	"FALSE"	Output divider0 interpolation phase shift enable;
CLK_DPS1_EN	string	"FALSE", "TRUE"	"FALSE"	Output divider1 interpolation phase shift enable;
CLK_DPS2_EN	string	"FALSE", "TRUE"	"FALSE"	Output divider2 interpolation phase shift enable;
CLK_DPS3_EN	string	"FALSE", "TRUE"	"FALSE"	Output divider3 interpolation phase shift enable;
CLK_DPS4_EN	string	"FALSE", "TRUE"	"FALSE"	Output divider4 interpolation phase shift enable;
CLK_DPS5_EN	string	"FALSE", "TRUE"	"FALSE"	Output divider5 interpolation phase shift enable;
CLK_DPS6_EN	string	"FALSE", "TRUE"	"FALSE"	Output divider6 interpolation phase shift enable;
CLK_DPSF_EN	string	"FALSE", "TRUE"	"FALSE"	Feedback F divider interpolation phase shift enable;
CLK_CAS5_EN	string	"FALSE", "TRUE"	"FALSE"	Output divider5 clock cascade enable;
CLKOUT0_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT0_SYN signal enable configuration;
CLKOUT1_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT1_SYN signal enable configuration;
CLKOUT2_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT2_SYN signal enable configuration;
CLKOUT3_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT3_SYN signal enable configuration;
CLKOUT4_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT4_SYN signal enable configuration;
CLKOUT5_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT5_SYN signal enable configuration;
CLKOUT6_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT6_SYN signal enable configuration;
CLKOUTF_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUTF_SYN signal enable configuration;
SSC_MODE	string	"DOWN_LOW", "DOWN_HIGH", "CENTER_LOW", "CENTER_HIGH", "DISABLE"	"DISABLE"	SSC mode configuration;
SSC_FREQ	real	25~250	50.0	SSC modulation frequency configuration, KHz;
INTERNAL_FB	string	"CLKOUT0", "CLKOUT1", "CLKOUT2", "CLKOUT3", "CLKOUT4", "CLKOUT5", "CLKOUT6", "CLKOUTF", "DISABLE"	"CLKOUTF"	Internal feedback path selection;
EXTERNAL_FB	string	"CLKOUT0", "CLKOUT1", "CLKOUT2", "CLKOUT3", "CLKOUT4", "CLKOUT5" "CLKOUT6", "CLKOUTF", "DISABLE"	"DISABLE"	External feedback path selection;

Parameter Name	Parameter Type	Settings	Defaults	Functional description
BANDWIDTH	string	"LOW", "HIGH" "OPTIMIZED"	"OPTIMIZED"	Bandwidth selection configuration;

Note: (1) Parameters CLK_DPS5_EN and CLK_CAS5_EN cannot be configured as "TRUE" at the same time;

(2) The parameters INTERNAL_FB and EXTERNAL_FB cannot be configured as "DISABLE" at the same time.

5. Instantiation template

```
GTP_GPLL #(
.CLKIN_FREQ(50),
.LOCK_MODE(0),
.STATIC_RATIOI(1),
.STATIC_RATIOM(1),
.STATIC_RATIO0(1),
.STATIC_RATIO1(1),
.STATIC_RATIO2(1),
.STATIC_RATIO3(1),
.STATIC_RATIO4(1),
.STATIC_RATIO5(1),
.STATIC_RATIO6(1),
.STATIC_RATIOF(1),
.STATIC_DUTY0(2),
.STATIC_DUTY1(2),
.STATIC_DUTY2(2),
.STATIC_DUTY3(2),
.STATIC_DUTY4(2),
.STATIC_DUTY5(2),
.STATIC_DUTY6(2),
.STATIC_DUTYF(2),
.STATIC_PHASE(0),
.STATIC_PHASE0(0),
.STATIC_PHASE1(0),
.STATIC_PHASE2(0),
.STATIC_PHASE3(0),
.STATIC_PHASE4(0),
.STATIC_PHASE5(0),
.STATIC_PHASE6(0),
.STATIC_PHASEF(0),
.STATIC_CPHASE0(0),
.STATIC_CPHASE1(0),
.STATIC_CPHASE2(0),
.STATIC_CPHASE3(0),
.STATIC_CPHASE4(0),
.STATIC_CPHASE5(0),
.STATIC_CPHASE6(0),
.STATIC_CPHASEF(0),
```

```
.CLK_DPS0_EN("FALSE"),
.CLK_DPS1_EN("FALSE"),
.CLK_DPS2_EN("FALSE"),
.CLK_DPS3_EN("FALSE"),
.CLK_DPS4_EN("FALSE"),
.CLK_DPS5_EN("FALSE"),
.CLK_DPS6_EN("FALSE"),
.CLK_DPSF_EN("FALSE"),
.CLK_CAS5_EN("FALSE"),
.CLKOUT0_SYN_EN("FALSE"),
.CLKOUT1_SYN_EN("FALSE"),
.CLKOUT2_SYN_EN("FALSE"),
.CLKOUT3_SYN_EN("FALSE"),
.CLKOUT4_SYN_EN("FALSE"),
.CLKOUT5_SYN_EN("FALSE"),
.CLKOUT6_SYN_EN("FALSE"),
.CLKOUTF_SYN_EN("FALSE"),
.SSC_MODE("DISABLE"),
.SSC_FREQ(50),
.INTERNAL_FB("CLKOUTF"),
.EXTERNAL_FB("DISABLE"),
.BANDWIDTH("OPTIMIZED")
) GTP_GPLL_inst (
.APB_RDATA    (apb_rdata),
.APB_ADDR     (apb_addr ),
.APB_WDATA    (apb_wdata),
.APB_READY    (apb_ready),
.CLKOUT0      (clkout0  ),
.CLKOUT0N     (clkout0n ),
.CLKOUT1      (clkout1  ),
.CLKOUT1N     (clkout1n ),
.CLKOUT2      (clkout2  ),
.CLKOUT2N     (clkout2n ),
.CLKOUT3      (clkout3  ),
.CLKOUT3N     (clkout3n ),
.CLKOUT4      (clkout4  ),
.CLKOUT5      (clkout5  ),
.CLKOUT6      (clkout6  ),
.CLKOUTF      (clkoutf  ),
.CLKOUTFN     (clkoutfn ),
.DPS_DONE     (dps_done ),
.LOCK         (lock     ),
.APB_CLK      (apb_clk  ),
.APB_EN       (apb_en   ),
.APB_RST_N    (apb_rst_n),
.APB_SEL      (apb_sel  ),
.APB_WRITE    (apb_write),
.CLKFB        (clkfb    ),
```

```
.CLKIN1        (clkin1    ),
.CLKIN2        (clkin2    ),
.CLKIN_SEL     (clkin_sel ),
.CLKOUT0_SYN   (clkout0_syn),
.CLKOUT1_SYN   (clkout1_syn),
.CLKOUT2_SYN   (clkout2_syn),
.CLKOUT3_SYN   (clkout3_syn),
.CLKOUT4_SYN   (clkout4_syn),
.CLKOUT5_SYN   (clkout5_syn),
.CLKOUT6_SYN   (clkout6_syn),
.CLKOUTF_SYN   (clkoutf_syn),
.DPS_CLK       (dps_clk   ),
.DPS_DIR       (dps_dir   ),
.DPS_EN        (dps_en    ),
.PLL_PWD       (pll_pwd   ),
.RST           (rst       )
);
```

6. Detailed function introduction

For detailed function introduction, please refer to "UG050004_Titan2 Series FPGA Clock Resource (Clock) User Guide".

(12) GTP_PPLL instruction

1. Supported devices

Table 287 GTP_PPLL device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

The structure block diagram of GTP_PPLL is shown in the figure below.

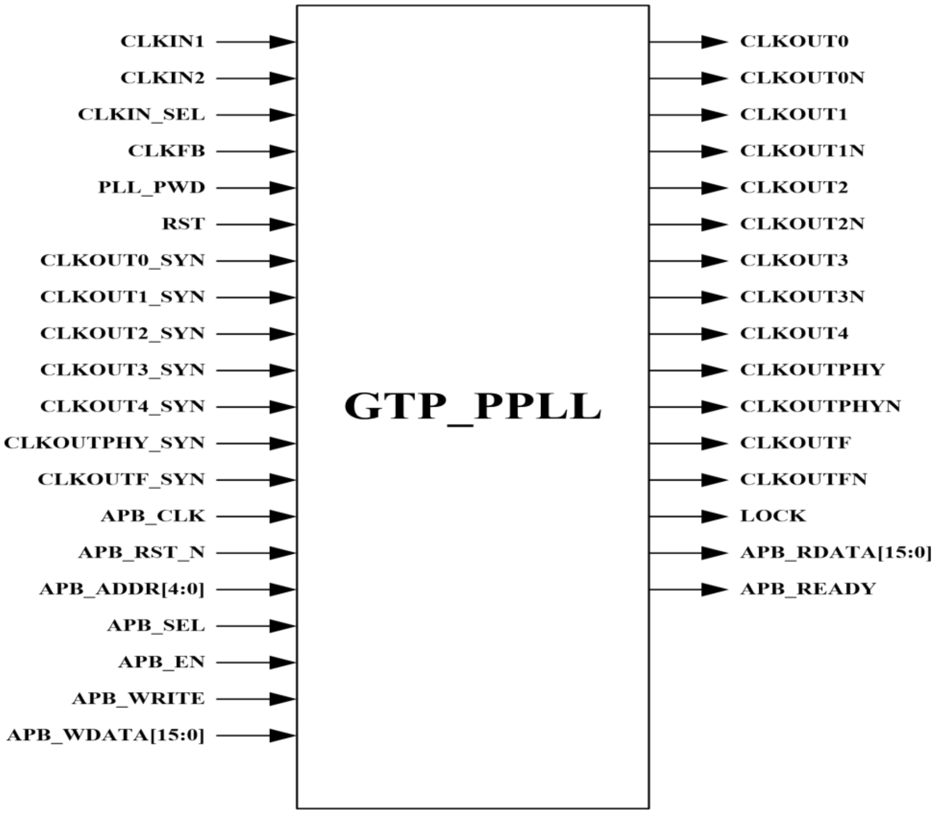


Figure 104 GTP_PPLL structure diagram

3. Port description

Table 288 GTP_PPLL port description

Port signal	Input / Output	Describe
CLKOUT0	Output	The positive phase output clock of the 0th channel of PLL;
CLKOUT0N	Output	PLL channel 0 inverted output clock;
CLKOUT1	Output	PLL No. 1 non-inverting output clock;
CLKOUT1N	Output	PLL 1st channel inverted output clock;
CLKOUT2	Output	PLL No. 2 non-inverting output clock;
CLKOUT2N	Output	PLL No. 2 inverted output clock;
CLKOUT3	Output	PLL No. 3 positive phase output clock;
CLKOUT3N	Output	PLL No. 3 inverted output clock;
CLKOUT4	Output	PLL No. 4 positive phase output clock;
CLKOUTPHY	Output	PLL to DDR PHY positive phase output clock;
CLKOUTPHYN	Output	PLL inverts output clock to DDR PHY;
CLKOUTF	Output	PLL feedback positive phase output clock;
CLKOUTFN	Output	PLL feedback inverting output clock;
LOCK	Output	PLL frequency lock indication signal, which is an asynchronous signal; when the signal is pulled high, it indicates that the PLL feedback clock signal has been locked to the input clock signal;
APB_RDATA[15:0]	Output	PLL APB interface data bus data output;
APB_READY	Output	PLL APB interface data bus handshake signal; indicates the end of a normal bus cycle;
CLKIN1	Input	PLL reference input clock 1;
CLKIN2	Input	PLL reference input clock 2;
CLKFB	Input	PLL feedback clock;
CLKIN_SEL	Input	input clock selection signal;
PLL_PWD	Input	PLL Power Down; active high;

Port signal	Input / Output	Describe
RST	Input	PLL reset signal, active high;
CLKOUT0_SYN	Input	CLKOUT0/N output clock enable control; high level active; high level off, low level enable;
CLKOUT1_SYN	Input	CLKOUT1/N output clock enable control; high level active; high level off, low level enable;
CLKOUT2_SYN	Input	CLKOUT2/N output clock enable control; high level active; high level off, low level enable;
CLKOUT3_SYN	Input	CLKOUT3/N output clock enable control; high level active; high level off, low level enable;
CLKOUT4_SYN	Input	CLKOUT4 output clock enable control; high level active; high level off, low level enable;
CLKOUTPHY_SYN	Input	CLKOUTPHY Output clock enable control; high level active; high level off, low level enable;
CLKOUTF_SYN	Input	CLKOUTF/N output clock enable control; high level active; high level off, low level enable;
APB_CLK	Input	PLL APB interface data bus clock;
APB_RST_N	Input	PLL APB interface data bus asynchronous reset signal, active low;
APB_ADDR	Input	PLL APB interface data bus address;
APB_SEL	Input	PLL APB interface data bus selection signal, select slave device;
APB_EN	Input	PLL APB interface data bus enable signal; indicates the second and subsequent cycles of the transfer;
APB_WRITE	Input	PLL APB interface data bus write enable signal; 1'b0: read operation, 1'b1: write operation;
APB_WDATA[15:0]	Input	PLL APB interface data bus data input;

4. Parameter description

Table 289 GTP_PPLL parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
CLKIN_FREQ	real	19~800	50.0	Input clock frequency configuration, MHz;
LOCK_MODE	binary	1'b0, 1'b1	1'b0	PLL frequency detection mode configuration; 1'b0: Real-time monitoring of PLL working status; 1'b1: LOCK is kept after PLL is locked, unless reset or Power Down;
STATIC_RATIOI	interger	1-42	1	Enter divider ratio static configuration;
STATIC_RATIOM	interger	1-128	1	Feedback M divider ratio static configuration;
STATIC_RATIO0	interger	1-128	1	Output divider0 ratio static configuration;
STATIC_RATIO1	interger	1-128	1	Output divider1 ratio static configuration;
STATIC_RATIO2	interger	1-128	1	Output divider2 ratio static configuration;
STATIC_RATIO3	interger	1-128	1	Output divider3 ratio static configuration;
STATIC_RATIO4	interger	1-128	1	Output divider4 ratio static configuration;
STATIC_RATIOPHY	interger	1-128	1	Output dividerphy ratio static configuration;
STATIC_RATIOF	interger	1-128	1	Feedback F divider ratio static configuration;
STATIC_DUTY0	interger	2~255	2	Output divider0 duty static configuration;
STATIC_DUTY1	interger	2~255	2	Output divider1 duty static configuration;
STATIC_DUTY2	interger	2~255	2	Output divider2 duty static configuration;
STATIC_DUTY3	interger	2~255	2	Output divider3 duty static configuration;
STATIC_DUTY4	interger	2~255	2	Output divider4 duty static configuration;

Parameter Name	Parameter Type	Settings	Defaults	Functional description
STATIC_DUTYPHY	interger	2~255	2	Output dividerphy duty static configuration;
STATIC_DUTYF	interger	2~255	2	Feedback F divider duty static configuration;
STATIC_PHASE0	interger	0-7	0	Output divider0 fine phase static configuration;
STATIC_PHASE1	interger	0-7	0	Output divider1 fine phase static configuration;
STATIC_PHASE2	interger	0-7	0	Output divider2 fine phase static configuration;
STATIC_PHASE3	interger	0-7	0	Output divider3 fine phase static configuration;
STATIC_PHASE4	interger	0-7	0	Output divider4 fine phase static configuration;
STATIC_PHASEPHY	interger	0-7	0	Output dividerphy fine phase static configuration;
STATIC_PHASEF	interger	0-7	0	Feedback F divider fine phase static configuration;
STATIC_CPHASE0	interger	0-127	0	Output divider0 coarse phase static configuration;
STATIC_CPHASE1	interger	0-127	0	Output divider1 coarse phase static configuration;
STATIC_CPHASE2	interger	0-127	0	Output divider2 coarse phase static configuration;
STATIC_CPHASE3	interger	0-127	0	Output divider3 coarse phase static configuration;
STATIC_CPHASE4	interger	0-127	0	Output divider4 coarse phase static configuration;
STATIC_CPHASEPHY	interger	0-127	0	Output dividerphy coarse phase static configuration;
STATIC_CPHASEF	interger	0-127	0	Feedback F divider coarse phase static configuration;
CLKOUT0_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT0_SYN signal enable configuration;
CLKOUT1_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT1_SYN signal enable configuration;
CLKOUT2_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT2_SYN signal enable configuration;
CLKOUT3_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT3_SYN signal enable configuration;
CLKOUT4_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUT4_SYN signal enable configuration;
CLKOUTPHY_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUTPHY_SYN signal enable configuration;
CLKOUTF_SYN_EN	string	"FALSE", "TRUE"	"FALSE"	CLKOUTF_SYN signal enable configuration;
INTERNAL_FB	string	"CLKOUT0", "CLKOUT1", "CLKOUT2", "CLKOUT3", "CLKOUT4", "CLKOUTF", "DISABLE"	"CLKOUTF"	Internal Feedback Path Selection;

Parameter Name	Parameter Type	Settings	Defaults	Functional description
EXTERNAL_FB	string	"CLKOUT0", "CLKOUT1" "CLKOUT2", "CLKOUT3", "CLKOUT4", "CLKOUTF", "DISABLE"	"DISABLE"	External feedback path selection;;
BANDWIDTH	string	"LOW", "HIGH" "OPTIMIZED"	"OPTIMIZED"	Bandwidth selection configuration;

Note: The parameters INTERNAL_FB and EXTERNAL_FB cannot be configured as "DISABLE" at the same time.

5. Instantiation template

```
GTP_PPLL #(
.CLKIN_FREQ(50),
.LOCK_MODE(0),
.STATIC_RATIOI(1),
.STATIC_RATIOM(1),
.STATIC_RATIO0(1),
.STATIC_RATIO1(1),
.STATIC_RATIO2(1),
.STATIC_RATIO3(1),
.STATIC_RATIO4(1),
.STATIC_RATIOPHY(1),
.STATIC_RATIOF(1),
.STATIC_DUTY0(2),
.STATIC_DUTY1(2),
.STATIC_DUTY2(2),
.STATIC_DUTY3(2),
.STATIC_DUTY4(2),
.STATIC_DUTYPHY(2),
.STATIC_DUTYF(2),
.STATIC_PHASE0(0),
.STATIC_PHASE1(0),
.STATIC_PHASE2(0),
.STATIC_PHASE3(0),
.STATIC_PHASE4(0),
.STATIC_PHASEPHY(0),
.STATIC_PHASEF(0),
.STATIC_CPHASE0(0),
.STATIC_CPHASE1(0),
.STATIC_CPHASE2(0),
.STATIC_CPHASE3(0),
.STATIC_CPHASE4(0),
.STATIC_CPHASEPHY(0),
.STATIC_CPHASEF(0),
.CLKOUT0_SYN_EN("FALSE"),
```

```
.CLKOUT1_SYN_EN("FALSE"),
.CLKOUT2_SYN_EN("FALSE"),
.CLKOUT3_SYN_EN("FALSE"),
.CLKOUT4_SYN_EN("FALSE"),
.CLKOUTPHY_SYN_EN("FALSE"),
.CLKOUTF_SYN_EN("FALSE"),
.INTERNAL_FB("CLKOUTF"),
.EXTERNAL_FB("DISABLE"),
.BANDWIDTH("OPTIMIZED")
) GTP_PPLL_inst (
.APB_RDATA    (apb_rdata         ),
.APB_ADDR     (apb_addr          ),
.APB_WDATA    (apb_wdata         ),
.APB_READY    (apb_ready         ),
.CLKOUT0      (clkout0           ),
.CLKOUT0N     (clkout0n          ),
.CLKOUT1      (clkout1           ),
.CLKOUT1N     (clkout1n          ),
.CLKOUT2      (clkout2           ),
.CLKOUT2N     (clkout2n          ),
.CLKOUT3      (clkout3           ),
.CLKOUT3N     (clkout3n          ),
.CLKOUT4      (clkout4           ),
.CLKOUT4N     (clkout4n          ),
.CLKOUTF      (clkoutf           ),
.CLKOUTFN     (clkoutfn          ),
.CLKOUTPHY    (clkoutphy         ),
.CLKOUTPHYN   (clkoutphyn        ),
.LOCK         (lock              ),
.APB_CLK      (apb_clk           ),
.APB_EN       (apb_en            ),
.APB_RST_N    (apb_rst_n         ),
.APB_SEL      (apb_sel           ),
.APB_WRITE    (apb_write         ),
.CLKFB        (clkfb             ),
.CLKIN1       (clkin1            ),
.CLKIN2       (clkin2            ),
.CLKIN_SEL    (clkin_sel         ),
.CLKOUT0_SYN  (clkout0_syn       ),
.CLKOUT1_SYN  (clkout1_syn       ),
.CLKOUT2_SYN  (clkout2_syn       ),
.CLKOUT3_SYN  (clkout3_syn       ),
.CLKOUT4_SYN  (clkout4_syn       ),
.CLKOUTF_SYN  (clkoutf_syn       ),
.CLKOUTPHY_SYN(clkoutphy_syn     ),
.PLL_PWD      (pll_pwd           ),
.RST          (rst               )
);
```

6. Detailed function introduction

For detailed function introduction, please refer to "UG050004_Titan2 Series FPGA Clock Resource (Clock) User Guide".

(13) GTP_DLL_E2 instruction

1. Supported devices

Table 290 GTP_DLL_E2 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

The main function of GTP_DLL_E2 is to dynamically lock the frequency of the input reference clock and output the equivalent number of delay steps. Its structural block diagram is shown below.

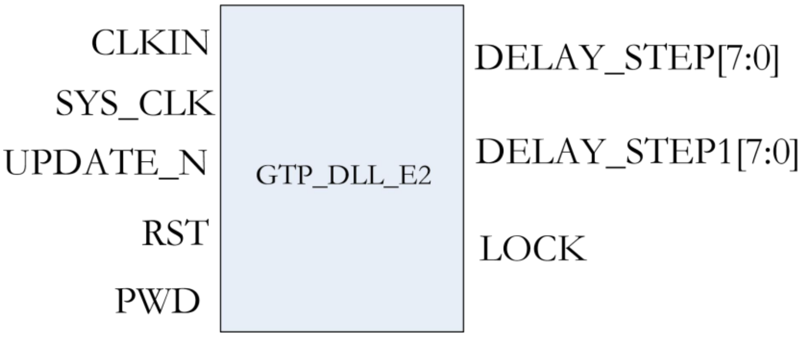


Figure 105 GTP_DLL_E2 structure diagram

3. Port description

Table 291 GTP_DLL_E2 port description

Port signal	Input / Output	Describe
CLKIN	Input	from IOCLK tree, fast clock
SYS_CLK	Input	From PPLL/SRB, DLL FSM clock
UPDATE_N	Input	Request to update the delay step of the DLL, active low
RST	Input	Active high reset signal
PWD	Input	Active high power down signal
DELAY_STEP	Output	90-degree delay control code output by DLL
DELAY_STEP1	Output	45 degree delay control code output by DLL
LOCK	Output	LOCK flag signal, high effective means LOCK

4. Parameter description

Table 292 GTP_DLL_E2 parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	<"TRUE", "FALSE">	<"TRUE">	Global reset enable signal
CAL_INIT	<interger>	<8'b00000000 ~ 8'b11111111>	<8'b00011111>	Calibration code
FAST_LOCK	<string>	<"TRUE", "FALSE">	<"FALSE">	DLL LOCK mode selection

DELAY_STEP_OFFSET	<interger>	<-4, -3, -2, -1, 0, 1, 2, 3, 4>	<0>	DLL output code code phase shift control
DELAY_SEL	<interger>	<0, 1>	<0>	DLL delay chain
FDIV	<interger>	<2'b00> <2'b01> <2'b10> <2'b11>	<2'b10>	Frequency division configuration, in DDR2/3 and QDR applications,CP_FDIV can only be set to 2'b10.
INT_CLK	<interger>	<0, 1>	<0>	DLL FSM SYSCLK source
UPD_DLY	<interger>	2'b00: 2 ioclk_2x cycles 2'b01: 4 ioclk_2x cycles 2'b10: 8 ioclk_2x cycles 2'b11: 16 ioclk_2x cycles	<2'b01>	DLL FSM calibration code dly_cal_therm_dly<63:0> update delay
HPIO	<string>	<"TRUE", "FALSE">	<"FALSE">	"FALSE" HRIO, "TRUE" HPIO

5. Instantiation template

```
GTP_DLL_E2 #(
.GRS_EN("TRUE"),
.CAL_INIT('b00011111),
.DELAY_STEP_OFFSET(0),
.DELAY_SEL(1'b0),
.FAST_LOCK("FALSE"),
.FDIV('b10),
.INT_CLK(1'b0),
.UPD_DLY('b01),
.HPIO("FALSE")
) GTP_DLL_E2_inst (
.DELAY_STEP(), // OUTPUT[7:0]
.DELAY_STEP1(),// OUTPUT[7:0]
.LOCK(),       // OUTPUT
.CLKIN(),      // INPUT
.PWD(),        // INPUT
.RST(),        // INPUT
.SYS_CLK(),    // INPUT
.UPDATE_N()    // INPUT
);
```

(14) GTP_OSC_E4 instruction

1. Supported devices

Table 293 GTP_OSC_E4 device support

Device family	Titan2

Whether to support the GTP	Support

2. Function introduction

When EN_N = 1, CLKOUT outputs low level; when EN_N = 0, CLKOUT outputs a fixed 50MHz clock to CCS as the system clock, the block diagram is shown below.

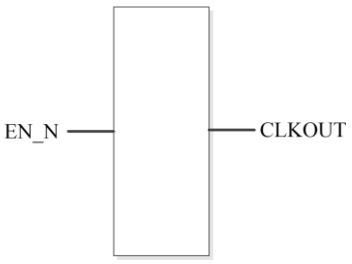


Figure 106 GTP_OSC_E4 structure diagram

3. Port description

Table 294 GTP_OSC_E4 port list

Port Name	Direction	Functional description
EN_N	Input	Output clock enable, active low
CLKOUT	Output	Output 50MHz clock

4. Instantiation template

```
GTP_OSC_E4 GTP_OSC_E4_inst (
.CLKOUT(clkout),
.EN_N(en_n)
);
```

7. Configuration related GTP instructions

(1) GTP_EFUSECODE usage instructions

1. Supported devices

Table 295 GTP_EFUSECODE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

The GTP is used to read EFUSECODE. Parallel output 32bit data stored in efuse to the user.

3. Port description

Table 296 GTP_EFUSECODE port list

Port Name	Direction	Functional description
EFUSE_CODE[31:0]	Output	Efusecode output data bus

4. Parameter description

Table 297 GTP_EFUSECODE parameter list

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
SIM_EFUSE_VALUE	Binary	0~32'hffffffff ;	32'h12345678	Efusecode value

5. Instantiation template

```
GTP_EFUSECODE#(
.SIM_EFUSE_VALUE       (32'h12345678)
) GTP_EFUSECODE_inst(
.EFUSE_CODE   (efuse   )
);
```

(2) GTP_IPAL_E2 instruction

1. Supported devices

Table 298 GTP_IPAL_E2 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IPAL_E2 is an internal slave parallel interface function, which is used to implement CRC detection or SEU detection on the read-back data. When writing a frame of data to the configuration memory, you must first write the high bit, and then write the low bit. So when reading back, It reads the high bit first, then the low bit. In X16 mode, a 32-bit word reads the high 16 bits first, and then reads the low 16 bits. The same is true for X8 mode.

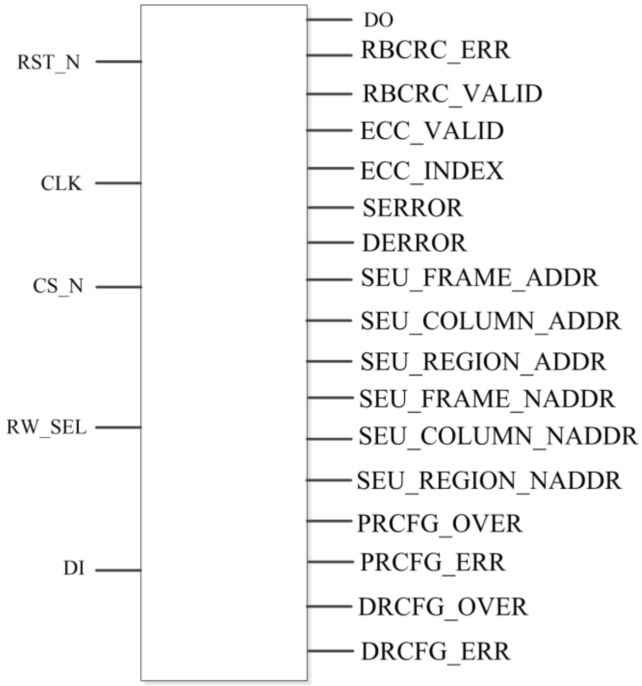


Figure 107 GTP_IPAL_E2 structure diagram

Users can instantiate the GTP to operate the internal slave parallel interface, or use the readback CRC and SEU interface to easily perform readback CRC and SEU operations. Here we mainly introduce the internal slave parallel interface (the RW_SEL signal transition can only performed while CS_N is high).

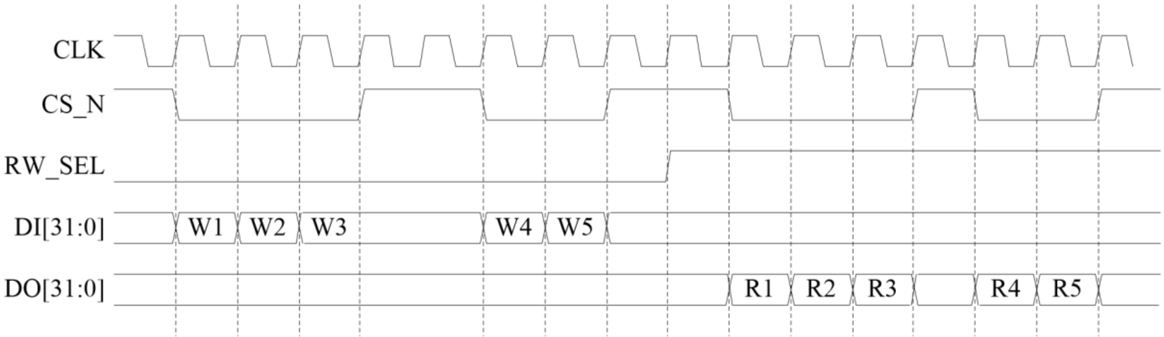


Figure 108 GTP_IPAL_E2 internal read and write interface timing from parallel interface

Table 299 GTP_IPAL_E2 emulation supported configuration registers

Name	Direction	Address	Functional description
IDR	R/W	00001	IDCODE register
CMDR	R/W	00010	Command register
CMEMIR	W	00101	Frame Data Input Register

Name	Direction	Address	Functional description
CMEMOR	R	00111	Frame Data Output Register
ADRR	R/W	01011	frame address register
SEUR	R/W	01101	SEU Control Register
SEUSTATUSR	R	01110	SEU Status Register
SEUADDR	R	11101	SEU Frame Address Register
SEUNADDR	R	11111	SEU next frame address register

3. Port description

Table 300 GTP_IPAL_E2 port list

Port Name	Direction	Functional description
RST_N	Input	Reset, active low
CLK	Input	Internal Parallel Interface Clock
CS_N	Input	Chip select signal, active low
RW_SEL	Input	Read and write options: 0 for writing, 1 for reading
DI[31:0]	Input	Data input
DO[31:0]	Output	Data output
RBCRC_ERR	Output	Read back CRC error flag
RBCRC_VALID	Output	Read back CRC valid flag
ECC_VALID	Output	ECC valid flag
ECC_INDEX[11:0]	Output	Address Index for Single Bit Errors
SERROR	Output	Single bit error flag for SEU detection
DERROR	Output	Double bit error flag for SEU detection
SEU_FRAME_ADDR[7:0]	Output	Current frame address detected by SEU
SEU_COLUMN_ADDR[7:0]	Output	Current column address for SEU detection
SEU_REGION_ADDR[4:0]	Output	Current zone address for SEU detection
SEU_FRAME_NADDR[7:0]	Output	Next frame address of SEU detection
SEU_COLUMN_NADDR[7:0]	Output	Next column address for SEU detection
SEU_REGION_NADDR[4:0]	Output	The address of the next area detected by the SEU
PRCFG_OVER	Output	Partial reconfiguration end flag
PRCFG_ERR	Output	Partial reconfiguration error flag
DRCFG_OVER	Output	Dynamic reconfiguration end flag
DRCFG_ERR	Output	Dynamic Reconfiguration Error Flags

4. Parameter description

Table 301 GTP_IPAL_E2 parameter list

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
IDCODE	<binary>	32'h0~32'hFFFF_FFFF	32'haaaa5555	Device IDCODE
DATA_WIDTH	<string>	"X8" "X16" "X32"	"X8"	Bit width selection of input data
SIM_DEVICE	<string>	"PG2L100H" "PG2T390H"	"PG2L100H"	Device Type Selection

5. Instantiation template

```
GTP_IPAL_E2 #(
.SIM_DEVICE("PG2L100H"),
```

```
.DATA_WIDTH("X8"),
.IDCODE('b10101010101010100101010101010101)
) GTP_IPAL_E2_inst (
.DO                    (do                ),
.ECC_INDEX             (ecc_index         ),
.SEU_COLUMN_ADDR       (seu_column_addr   ),
.SEU_COLUMN_NADDR      (seu_column_naddr  ),
.SEU_FRAME_ADDR        (seu_frame_addr    ),
.SEU_FRAME_NADDR       (seu_frame_naddr   ),
.SEU_REGION_ADDR       (seu_region_addr   ),
.SEU_REGION_NADDR      (seu_region_naddr  ),
.DI                    (di                ),
.DERROR                (derror            ),
.ECC_VALID             (ecc_valid         ),
.PRCFG_ERR             (prcfg_err         ),
.PRCFG_OVER            (prcfg_over        ),
.RBCRC_ERR             (rbcrc_err         ),
.RBCRC_VALID           (rbcrc_valid       ),
.SERROR                (serror            ),
.CLK                   (clk               ),
.CS_N                  (cs_n              ),
.RST_N                 (rst_n             ),
.RW_SEL                (rw_sel            )
);
```

(3) GTP_SCANCHAIN_E1 instruction

1. Supported devices

Table 302 GTP_SCANCHAIN_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

Through the JTAG interface, read the value of the user data register.

3. Port description

Table 303 GTP_SCANCHAIN_E1 port list

Port Name	Direction	Functional description
TDI	Input	Jtag interface
TDO	Output	Jtag interface
TMS	Input	Jtag interface
TCK	Input	Jtag interface
RST	Output	Jtag soft reset output
CAPDR	Output	Jtag capturedr status indication

Port Name	Direction	Functional description
SHFTDR	Output	Shiftdr status indication
UPDR	Output	Updatedr status indicator
JCLK	Output	Shiftdr thought clock
FLG_USER	Output	user instruction
TDI_USER	Output	User register data input
TDO_USER	Input	User register data output
JRTI	Output	Run/test idle status indication
TCK_USER	Output	Tck to user
TMS_USER	Output	Tms to user

4. Parameter description

Table 304 GTP_SCANCHAIN_E1 parameter list

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
IDCODE	Binary	0~32'hffffffff ;	32'haaaa5555	IDCODE
CHAIN_NUM	Integer	1, 2, 3, 4	1	User DR Number

5. Instantiation template

```
GTP_SCANCHAIN #(
.IDCODE      (32'haaaa5555),
.CHAIN_NUM (1)
)
GTP_SCANCHAIN_inst(

.TCK         (tck     ),
.TDI         (tdi     ),
.TMS         (tms     ),
.TDO         (tdo     ),
.CAPDR       (capture ),
.JCLK        (jclk    ),
.RST         (rst     ),
.FLG_USER    (flg_user),
.SHFTDR      (shift   ),
.TDI_USER    (tdi_user),
.TMS_USER    (tms_user),
.JRTI        (jrti    ),
.UPDR        (update  ),
.TDO_USER    (tdo_user)
);
```

6. Detailed function description

The user can read the chip IDCODE; the value of multiple user logic data registers can be read and written separately through the JTAG interface.

(4) GTP_UDID instruction

1. Supported devices

Table 305 GTP_UDID device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

This GTP is used to read UDID CODE value.

UID length is 96 bits.

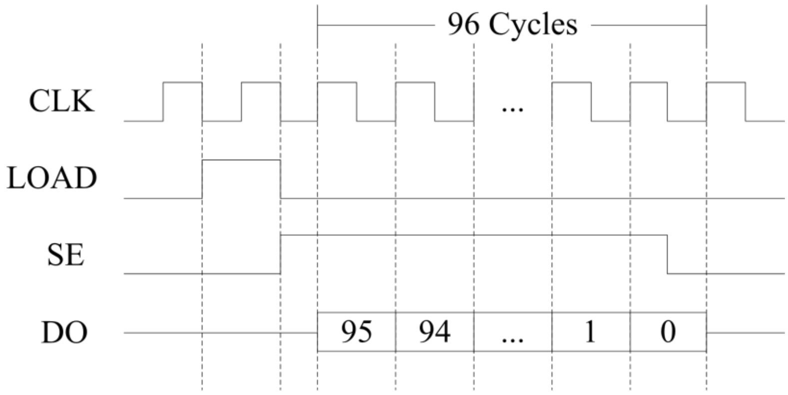


Figure 109 Read UID Timing

Users can extend the bit width of UID.

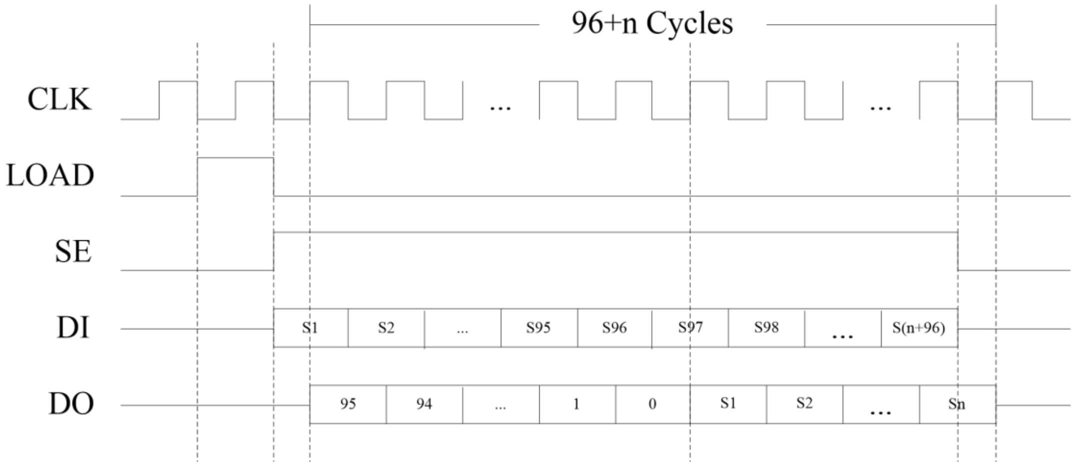


Figure 110 Extend UID bit width

3. Port description

Table 306 GTP_UDID port list

Port Name	Direction	Functional description
DI	Input	Serial data input
DO	Output	Serial data out
SE	Input	Data shift enable
LOAD	Input	Data registers load UID CODE in parallel
CLK	Input	Clock

4. Parameter description

Table 307 GTP_UDID parameter list

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
UDID_CODE	Binary	0~2^96 - 1; (all 0 to all 1)	0	Chip ID
UDID_WIDTH	integer	0~32'hffff_ffff	64	Set the bit width of UDID_CODE

Note: The above parameters are only used in the simulation test. Different series of devices have different UID lengths, and the default value is 64. Therefore, in order to keep the simulation consistent with the upper board, it should be changed to 96 for Titan2 devices.

5. Instantiation template

```
GTP_UDID#(
.UDID_WIDTH(64),
.UDID_CODE (0)
)
GTP_UDID_inst(
.DI          (di       ),
.DO          (do       ),
.LOAD        (load     ),
.SE          (se       ),
.CLK         (clk      )
);
```

6. Detailed function description

Serially output the chip identification code data to the user.

(5) GTP_JTAGIF instruction

1. Supported devices

Table 308 GTP_JTAGIF device support

Device family	Titan2
Whether to support the GTP	Support

2. Ability to introduce

GTP_JTAGIF has three input ports: TCK, TMS and TDI, one output port TDO, and one parameter USERCODE to realize the user JTAG interface function. Emulation only supports JTAG instruction operations, read and write configuration registers, and read and write configuration memory.

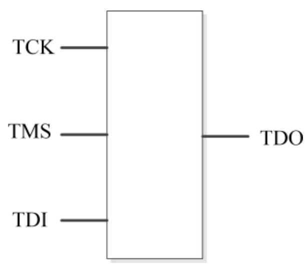


Figure 111 GTP_JTAGIF structure diagram

Table 309 GTP_JTAGIF emulation supported configuration registers

Name	Direction	Address	Functional description
IDR	R/W	00001	IDCODE register
CMDR	R/W	00010	Command register
CMEMIR	W	00101	Frame Data Input Register
CMEMOR	R	00111	Frame Data Output Register
ADRR	R/W	01011	Frame address register

3. Port description

Table 310 GTP_JTAGIF port list

Port Name	Direction	Functional description
TCK	Input	User JTAG Clock
TMS	Input	User JTAG test mode selection
TDI	Input	User JTAG test data input
TDO	Output	User JTAG test data output

4. Parameter description

Table 311 GTP_JTAGIF parameter list

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
USERCODE	binary	32'h0~32'hFFFF_FFFF	32'hFFFF_FFFF	USERCODE
IDCODE	binary	32'h0~32'hFFFF_FFFF	32'h5555_AAAA	IDCODE (provides emulation, but does not map)

5. Instantiation template

```
GTP_JTAGIF #(
.USERCODE(32'hFFFF_FFFF)
.IDCODE(32'h5555_AAAA)
) GTP_JTAGIF_inst (
.TDO(tdo),
.TCK(tck),
.TDI(tdi),
.TMS(tms)
);
```

(6) GTP_KEYRAM instruction

1. Supported devices

Table 312 GTP_KEYRAM device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

The function of GTP_KEYRAM is to clear the internal memory, and the structure diagram is as follows.

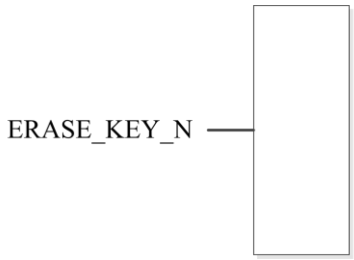


Figure 112 GTP_KEYRAM structure diagram

3. Port description

Table 313 GTP_KEYRAM port list

Port Name	Direction	Functional description
ERASE_KEY_N	Input	Cleared, active low.

4. Instantiation template

```
GTP_KEYRAM
GTP_KEYRAM_inst (
.ERASE_KEY_N (key_n)
);
```

(7) GTP_CFGCLK instruction

1. Supported devices

Table 314 GTP_CFGCLK device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_CFGCLK has only two inputs, a clock input signal and a clock enable signal. The GTP is the FLASH clock and clock enable sent to CCS by SRB.

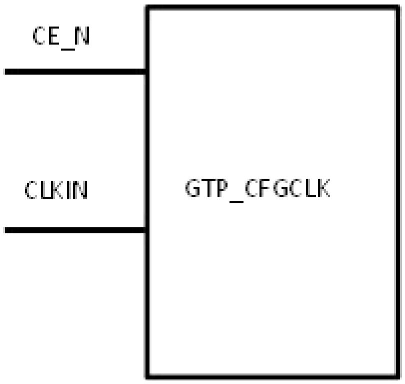


Figure 113 GTP_CFGCLK structure diagram

3. Port description

Table 315 GTP_CFGCLK port list

Port Name	Direction	Functional description
CLKIN	Input	User master clock: Used for SPI FLASH in-system direct programming and in-system indirect programming. Used for BPI FLASH in-system indirect programming.
CE_N	Input	User master clock enable, active low.

4. Instantiation template

```
GTP_CFGCLK GTP _CFGCLK_inst (
.CE_N         (ce_n), // INPUT
.CLKIN        (clkin) // INPUT
);
```

8. DDR Related GTP Instructions

(1) GTP_IOCLKDIV_E3 instruction

1. Supported devices

Table 316 GTP_IOCLKDIV_E3 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

The function of this GTP is to divide the clock from the PLL IOCLK clock tree to generate the system clock of DDRPHY.

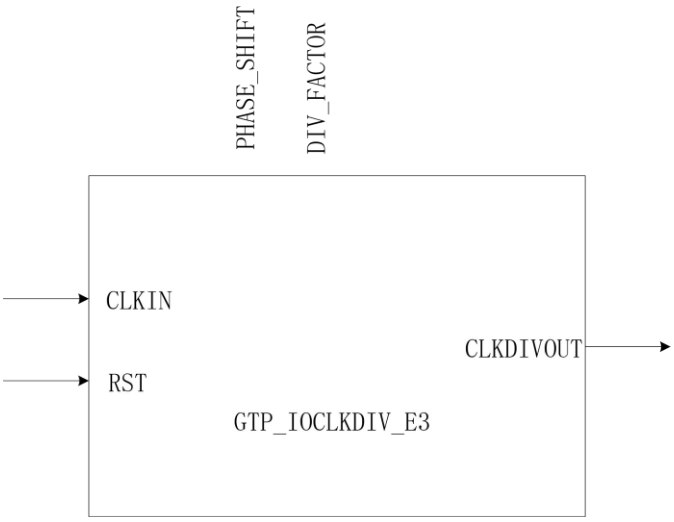


Figure 114 GTP_IOCLKDIV_E3 structure diagram

3. Port description

Table 317 GTP_IOCLKDIV_E3 port description

Port signal	Input / Output	Describe
CLKIN	Input	From PLL IOCLK clock tree
RST	Input	Local control signal, active high
CLKDIVOUT	Output	DDRPHY system clock

4. Parameter description

Table 318 GTP_IOCLKDIV_E3 parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
PHASE_SHIFT	<string>	"0" "1" "2" "3" "4" "5""6" "7"	<"0">	PHASE_SHIFT indicates how many CLKIN cycles CLKDIVOUT delays after RST is released. The value of PHASE_SHIFT must be smaller than the frequency division factor DIV_FACTOR.
DIV_FACTOR	<string>	"4" "8"	"8"	Frequency division factor

Parameter Name	Parameter Type	Settings	Defaults	Functional description
				"4": quarter frequency "8": eighth frequency

5. Instantiation template

```
GTP_IOCLKDIV_E3 #(
.DIV_FACTOR   ( "8"      ),
.PHASE_SHIFT( "0"       )
) GTP_IOCLKDIV_E3_inst (
.CLKDIVOUT    (clkdivout),
.CLKIN        (clkin     ),
.RST_N        (rst       )
);
```

(2) GTP_CLKPD instruction

1. Supported devices

Table 319 GTP_CLKPD device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

The main function of CLKPD is to detect the phase difference between MEMORY CONTROLLER soft core clock and DDR_PHY system clock, and output FLAG_PD signal to SRB. The soft core will convert the signal to DPS interface to adjust the output phase of GPLL. FLAG_PD will be between 0 and 1 jitter until the phase is locked, at this time LOCK becomes 1, which means that the soft core clock and the system clock are phase locked.

When the parameter HPIO="TRUE", CPD_EDGE can be selected as "POSEDGE" rising edge alignment and "NEGEDGE" falling edge alignment.

The structural block diagram is as follows:

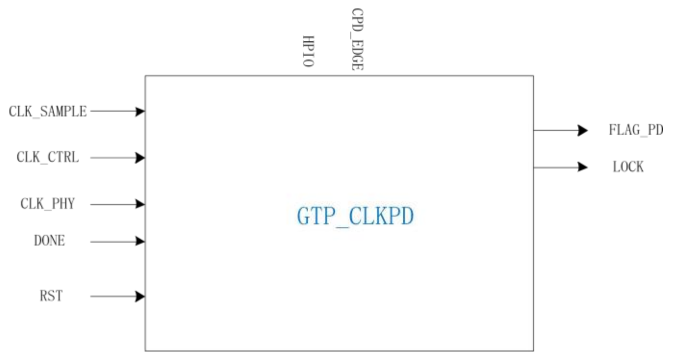


Figure 115 GTP_CLKPD structure diagram

3. Port description

Table 320 GTP_CLKPD port description

Port signal	Input / Output	Describe
RST	Input	CLKPD local reset control signal, active low
CLK_SAMPLE	Input	FLAG_PD sampling clock
CLK_CTRL	Input	MEMORY CONTROLLER soft core clock
CLK_PHY	Input	DDR_PHY system clock
DONE	Input	The CLKPD_DONE signal also becomes high level after LOCK changes from 0 to 1, and locks the CLKPD module.
FLAG_PD	Output	"1" increases the delay for the control GPLL; "0" decreases the delay for the control GPLL
LOCK	Output	Switch to high level when FLAG_PD is detected to be shaking between 0 and 1

4. Parameter description

Table 321 GTP_CLKPD parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
HPIO	string	"TRUE", "FALSE"	"FALSE"	"TRUE"HPIO, "FALSE" HRIO
CPD_EDGE	string	"POSEDGE", "NEGEDGE"	"POSEDGE"	"POSEDGE" rising edge alignment, "NEGEDGE" falling edge alignment

5. Instantiation template

```
GTP_CLKPD
gtp_cpd_a# (
.HPIO ("TRUE"),
.CPD_EDGE ("POSEDGE")
) (
.FLAG_PD (cpd_up_dnb),
.LOCK (cpd_lock),
.RST (cpd_rstn),
.CLK_SAMPLE (clk_io_2x),
.CLK_CTRL (clk_fb),
.CLK_PHY (pll_sysclk),
.DONE (cpd_done)
);
```

(3) GTP_DDC_E2 instruction

1. Supported devices

Table 322 GTP_DDC_E2 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

The main function of GTP_DDC_E2 is to generate write operation clock for DDR memory; support DDR3 write leveing process, in quad/half rate mode, support clock shutdown operation, and shutdown time is an intege multiple of

CLKA cycles; read The DQSI signal in operation is phase-shifted by ~90 degrees to capture the input data and generate the gate correct identification signal DGTS; perform drift detection on the DQS signal and send the detection result to the update logic for further processing; generate read and write for the input FIFO Point to the operation signal; and generate a valid judgment signal for the read operation for the core logic.

The structural block diagram is as follows:

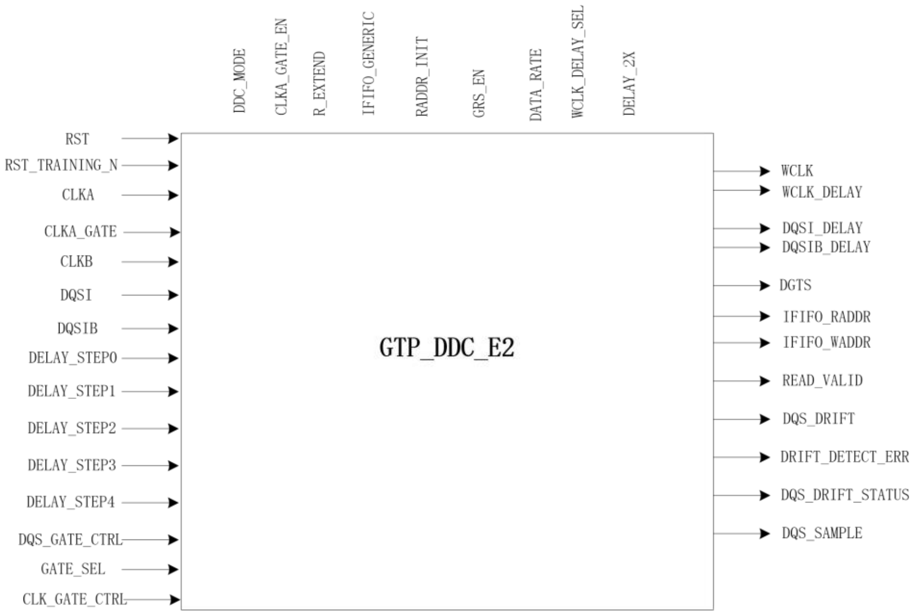


Figure 116 GTP_DDC_E2 structure diagram

3. Port description

Table 323 GTP_DDC_E2 port description

Port signal	Input / Output	Describe
RST	Input	DQS local reset control signal, the polarity is controlled by GRS_EN
RST_TRAINING_N	Input	Used to reset the reset signal of DQSL's read and write pointer generation and read pointer control logic at the end of dqs gate training, active low
CLKA	Input	The clock source is IOCLK tree
CLKA_GATE	Input	For gate clka clock
CLKB	Input	System clock clock from PPLL/SRB
DQSI	Input	Read clock signal
DQSIB	Input	Read clock signal
DELAY_STEP0[7:0]	Input	Code for write leveling training, controlled by soft calibration logic.
DELAY_STEP1[7:0]	Input	Code for Read DQSi even eye training
DELAY_STEP2[7:0]	Input	Code for Read DQSi odd eye training
DELAY_STEP3[7:0]	Input	45L delay control code, from DLL
DELAY_STEP4[7:0]	Input	Code for DQ/DQS training
DQS_GATE_CTRL[3:0]	Input	Used to control the timing of DQS gate
GATE_SEL	Input	DQS GATE Coarse adjustment selection signal, the signal selected when "1" is delayed by one clock cycle than the signal selected when "0"
CLK_GATE_CTRL[1:0]	Input	DQS GATE Fine-tuning control signal with 90° phase difference between each value
WCLK	Output	DQS Write clock
WCLK_DELAY	Output	Clock signal of WCLK phase shift 270K_CLK

Port signal	Input / Output	Describe
DQSI_DELAY	Output	DQSI signal ~90I phase-shifted signal to capture input data
DQSIB_DELAY	Output	DQSIB signal~90I phase-shifted signal to capture input data
DGTS	Output	DQS gate status identification, "1"gate is correct, "0"gate is wrong
DQS_DRIFT[1:0]	Output	DQS delay change detection signal, update logic records the signal and judges whether to perform gate window position update adjustment
DRIFT_DETECT_ERR	Output	DQS delay change detection error flag, high level triggers PHY to perform update training operation
DQS_DRIFT_STATUS	Output	DQS delay exceeds the range identification signal, high level triggers PHY to perform update training operation
READ_VALID	Output	Read IFIFO operation effective judgment signal, high level is effective
IFIFO_WADDR	Output	IFIFO write pointer, expressed in gray code
IFIFO_RADDR	Output	IFIFO read pointer, expressed in gray code
DQS_SAMPLE	Output	DQS gate sampling used by soft core training logic

4. Parameter description

Table 324 GTP_DDC_E2 parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
GRS_EN	<string>	<"TRUE", "FALSE">	<"TRUE">	"TRUE" global RST enabled, "FALSE" global RST disabled
DDC_MODE	<string>	<"HALF_RATE", "QUAD_RATE">	<"QUAD_RATE">	DQS working mode selection signal "HALF_RATE": x2 mode; support write leveling "QUAD_RATE": x4 mode; support write leveling
CLKA_GATE_EN	<string>	<"TRUE", "FALSE">	<"FALSE">	"TRUE": The GATE function is enabled, and CLKA is turned off, so that the entire DQS cannot work in QUAD/HALF_RATE mode; "FALSE": GATE function disable, do not shut down CLKA;
R_EXTEND	<string>	<"TRUE", "FALSE">	<"FALSE">	When the DQS gate window position control signal is "TRUE", the gate window position is delayed compared to "FALSE": "QUAD_RATE" is delayed by 1 wclk cycle; "HALF_RATE" is delayed by 1 wclk cycle;
IFIFO_GENERIC	<string>	<"TRUE", "FALSE">	<"FALSE">	Read and write fifo mode control signal "TRUE": GENERIC mode; "FALSE": DDR MEM mode;
RADDR_INIT	<bit>	<3'b000~3'b111>	<3'b000>	Read address initial value in Generic mode
WCLK_DELAY_SEL	<string>	<"TRUE", "FALSE">	<"FALSE">	CLK_W_DEL source selection signal "FALSE" selects ~dqsw90, and the phase shift between CLK_W and CLK_W is 270 degrees; "TRUE" selects dqsw90, and the phase shift between CLK_W and CLK_W is 90 degrees;
DELAY_SEL	<bit>	<1'b1, 1'b0>	<1'b0 >	Increase the delay of the delay chain for low-speed interfaces below 800MHz 1'b0:1x delay range; 1'b1: 2x delay range

Parameter Name	Parameter Type	Settings	Defaults	Functional description
DATA_RATE	<bit>	<2'b00, 2'b01 >	<2'b00>	Adjust the LDO output voltage according to different frequencies to meet the delay chain requirements 2'b00: 533-2133Mbps 2'b01: 400Mbps

5. Instantiation template

```
GTP_DDC_E2 #(
.GRS_EN("TRUE"),
.CLKA_GATE_EN("FALSE"),
.WCLK_DELAY_SEL("FALSE"),
.DDC_MODE("QUAD_RATE"),
.R_EXTEND("FALSE"),
.DELAY_SEL(1'b0),
.IFIFO_GENERIC("FALSE"),
.RADDR_INIT('b000),
.DATA_RATE('b00)
) GTP_DDC_E2_inst (
.DQS_DRIFT(),
.IFIFO_RADDR(),
.IFIFO_WADDR(),
.CLK_GATE_CTRL(),
.DELAY_STEP0(),
.DELAY_STEP1(),
.DELAY_STEP2(),
.DELAY_STEP3(),
.DELAY_STEP4(),
.DQS_GATE_CTRL(),
.DGTS(),
.DQSIB_DELAY(),
.DQSI_DELAY(),
.DQS_DRIFT_STATUS(),
.DQS_SAMPLE(),
.DRIFT_DETECT_ERR(),
.READ_VALID(),
.WCLK(),
.WCLK_DELAY(),
.CLKA(),
.CLKA_GATE(),
.CLKB(),
.DQSI(),
.DQSIB(),
.GATE_SEL(),
.RST(),
.RST_TRAINING_N()
);
```

(4) GTP_IDDR_E1 instruction

1. Supported devices

Table 325 GTP_IDDR_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_IDDR_E1 is a data deserializer, its structural block diagram is shown below, it supports IDDRx1, SAME_PIPELINED, SAME_EDGE and OPPOSITE_EDGE mode.

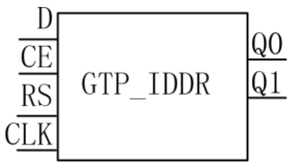


Figure 117 GTP_IDDR_E1 structure diagram

3. Port description

Table 326 GTP_IDDR_E1 port description

Port signal	Input / Output	Data width	Describe
D	Input	1	Data input
CE	Input	1	Clock enable
RS	Input	1	Set/reset
CLK	Input	1	System clock
Q0	Output	1	Deserialized output
Q1	Output	1	Deserialized output

4. Parameter description

Table 327 GTP_CLKBUFXCE parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
IDDR_MODE	string	"SAME_PIPELINED" "SAME_EDGE" "OPPOSITE_EDGE"	"OPPOSITE_EDGE"	"SAME_PIPLELINED": DDR 1:2 same pipelined deserialization mode "SAME_EDGE": DDR 1:2 SAME_EDGE deserialization mode "OPPOSITE_EDGE": DDR 1:2 OPPOSITE_EDGE deserialization mode
RS_TYPE	string	"SYNC_SET" "ASYNC_SET" "SYNC_RESET" "ASYNC_RESET"	"ASYNC_RESET"	Reset/set mode selection: synchronous set, asynchronous set Synchronous reset, asynchronous reset
GRS_EN	string	"FALSE" "TRUE"	"TRUE"	Global reset enable

5. Instantiation template

```
GTP_IDDR_E1 # (
. GRS_EN("TRUE"),
. IDDR_MODE("OPPOSITE_EDGE"),
. RS_TYPE("ASYNC_RESET")
) GTP_IDDR_E1_inst (
.D (d),
.CE(ce),
.RS(rs),
.CLK(clk),
.Q0(q0),
.Q1(q1)
);
```

(5) GTP_ODDR_E1 instruction

1. Supported devices

Table 328 GTP_ODDR_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_ODDR_E1 is a data parallel-to-serial processing module, which supports 2:1, SAME_EDGE, OPPOSITE_EDGE mode, and its structural block diagram is as follows.

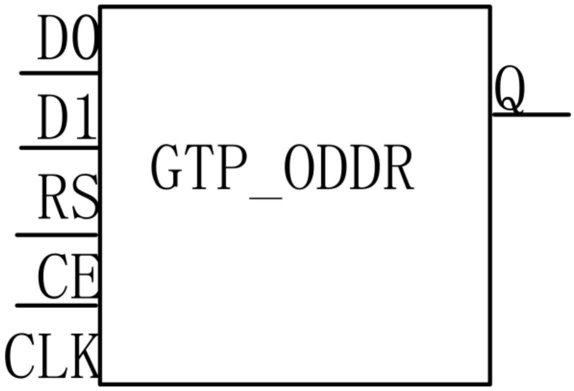


Figure 118 GTP_ODDR_E1 structure diagram

3. Port description

Table 329 GTP_ODDR_E1 port description

Port signal	Input / Output	Data width	Describe
D0	Input	1	Data input
D1	Input	1	Data input
CE	Input	1	Clock enable

Port signal	Input / Output	Data width	Describe
RS	Input	1	Set/reset
CLK	Input	1	System clock
Q	Output	1	Serial data out

4. Parameter description

Table 330 GTP_ODDR_E1 parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
ODDR_MODE	string	"SAME_EDGE" "OPPOSITE_EDGE"	"SAME_EDGE"	"SAME_EDGE": DDR 2:1 serialization mode "OPPOSITE_EDGE": DDR 2:1 serialization mode
GRS_EN	string	"FALSE" "TRUE"	"TRUE"	Global reset enable
RS_TYPE	string	"SYNC_SET" "ASYNC_SET" "SYNC_RESET" "ASYNC_RESET"	"ASYNC_RESET"	Reset/set mode selection: Synchronous set, Asynchronous set Synchronous reset, Asynchronous reset

5. Instantiation template

```
GTP_ODDR_E1 # (
. GRS_EN("TRUE"),
. ODDR_MODE("SAME_EDGE"),
. RS_TYPE ("ASYNC_RESET")
) GTP_ODDR_E1_inst (
.D0(d0),
.D1(d1),
.CE(ce),
.RS(rs),
.CLK(clk),
.Q (q)
);
```

9. Other related GTP instructions

(1) GTP_APM_E2 instruction

1. Supported devices

Table 331 GTP_APM_E2 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_APM_E2 is a logical operation unit that supports various types of logical operations, including multiplication, multiply-accumulate, general-purpose multiply-accumulate, wide-bit accumulation and addition, and FIR logic operations. The block diagram is as follows:

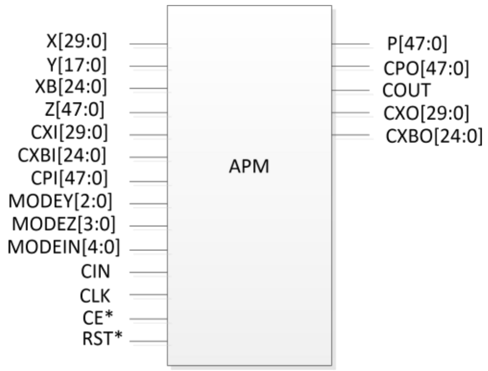


Figure 119 GTP_APM_E2 structure diagram

3. Port description

Table 332 GTP_APM_E2 port description

Port signal	Input / Output	Describe
X[29:0]	Input	Parallel data input X
CXI[29:0]	Input	Cascade X input
CXBI[24:0]	Input	Cascade H input
XB[24:0]	Input	Parallel Data Input XB
Y[17:0]	Input	Parallel data input Y
Z[47:0]	Input	Parallel data input Z
CPI[47:0]	Input	Cascade P input
CIN	Input	Cascade CIN input
MODEY[2:0]	Input	APM dynamic Y end control operator
MODEZ[3:0]	Input	APM dynamic Z terminal control operator
MODEIN[4:0]	Input	APM dynamic input control operator
CLK	Input	Clock input for DSP MU 0
CEX1	Input	Clock X REG1 enable input
CEX2	Input	Clock X REG2 enable input
CEX3	Input	Clock X REG3 enable input
RSTX	Input	High effective X Reset input
CEXB	Input	Clock XB enable input

Port signal	Input / Output	Describe
RSTXB	Input	High effective XB Reset input
CEY1	Input	Clock Y REG1 enable input
CEY2	Input	Clock Y REG2 enable input
RSTY	Input	High effective Y Reset input
CEZ	Input	Clock Z enable input
RSTZ	Input	High effective Z Reset input
CEPRE	Input	Clock PRE enable input
RSTPRE	Input	High effective PRE Reset input
CEM	Input	Clock M enable input
RSTM	Input	High effective M Reset input
CEP	Input	Clock P enable input
RSTP	Input	High effective P Reset input
CEMODEIN	Input	Clock MODEIN enable input
RSTMODEIN	Input	High effective MODEIN Reset input
CEMODEY	Input	Clock MODEY enable input
RSTMODEY	Input	High effective MODEY Reset input
CEMODEZ	Input	Clock MODEZ enable input
RSTMODEZ	Input	High effective MODEZ Reset input
P[47:0]	Output	Parallel data output
CPO[47:0]	Output	Cascade P output
COUT	Output	Cascade CIN output
CXO[29:0]	Output	Cascade X output
CXBO[24:0]	Output	Cascade XB output

4. Parameter description

Table 333 GTP_APM_E2 parameter list description

Parameter Name	Parameter Type	Settings	Defaults	Functional description
USE_POSTADD	int	0, 1	0	Post adder enable
USE_PREADD	int	0, 1	0	Pre-adder enable
CXO_REG	int	0, 1, 2, 3	0	X cascade out reg latency
X_REG	int	0, 1, 2, 3	0	X_REG register enable
XB_REG	int	0, 1	0	XB_REG register enable
Y_REG	int	0, 1, 2, 3	0	Y_REG register enable
Z_REG	int	0, 1	0	Z_REG register enable
PREADD_REG	int	0, 1	0	Preadd register enable
MULT_REG	int	0, 1	0	Mult register enable
P_REG	int	0, 1	0	Postadd register enable
MODEIN_REG	int	0, 1	0	MODEIN_REG register enable
MODEY_REG	int	0, 1	0	MODEY_REG register enable
MODEZ_REG	int	0, 1	0	MODEZ_REG register enable
X_SEL	int	0, 1	0	Mult X input select
XB_SEL	int	0, 1, 2, 3	0	X back propagate mux select
ASYNC_RST	int	0, 1	0	Async reset enable
USE_SIMD	int	0, 1	0	Mode selector
P_INIT0	binary	48'h0-48'hfff	48'h0	P constant input0

Parameter Name	Parameter Type	Settings	Defaults	Functional description
P_INIT1	binary	48'h0-48'hfff	48'h0	P constant input1
ROUNDMODE_SEL	int	0, 1	0	Roundmode selection
CPO_REG	int	0, 1	0	PO, PCO use register output enable
USE_ACCLOW	int	0, 1	0	Acc use lower 18-bit feedback only enable
CIN_SEL	int	0, 1	0	Select PCI for postadder carry in
GRS_EN	string	"TRUE", "FALSE"	"TRUE"	Global Reset enable

5. Instantiation template

```
GTP_APM_E2 #(
.GRS_EN("TRUE"),
.ASYNC_RST(0),
.X_REG(0),
.XB_REG(0),
.Y_REG(0),
.Z_REG(0),
.P_REG(0),
.CXO_REG(0),
.CPO_REG(0),
.MULT_REG(0),
.PREADD_REG(0),
.MODEIN_REG(0),
.MODEY_REG(0),
.MODEZ_REG(0),
.X_SEL(0),
.XB_SEL(0),
.CIN_SEL(0),
.ROUNDMODE_SEL(0),
.USE_SIMD(0),
.USE_ACCLOW(0),
.USE_PREADD(0),
.USE_POSTADD(0),
.P_INIT0('b000000000000000000000000000000000000000000000000),
.P_INIT1('b000000000000000000000000000000000000000000000000)
) GTP_APM_E2_inst (
.CPO          (cpo       ),
.CXBO         (cxbo      ),
.CXO          (cxo       ),
.P            (p         ),
.CPI          (cpi       ),
.CXBI         (cxbi      ),
.CXI          (cxi       ),
.MODEIN       (modein    ),
.MODEY        (modey     ),
.MODEZ        (modez     ),
.X            (x         ),
.XB           (xb        ),
```

```
    .Y             (y         ),
    .Z             (z         ),
    .COUT          (cout      ),
    .CEM           (cem       ),
    .CEMODEIN      (cemodein),
    .CEMODEY       (cemodey  ),
    .CEMODEZ       (cemodez  ),
    .CEP           (cep       ),
    .CEPRE         (cepre     ),
    .CEX1          (cex1      ),
    .CEX2          (cex2      ),
    .CEX3          (cex3      ),
    .CEXB          (cexb      ),
    .CEY1          (cey1      ),
    .CEY2          (cey2      ),
    .CEZ           (cez       ),
    .CIN           (cin       ),
    .CLK           (clk       ),
    .RSTM          (rstm      ),
    .RSTMODEIN     (rstmodein),
    .RSTMODEY      (rstmodey ),
    .RSTMODEZ      (rstmodez ),
    .RSTP          (rstp      ),
    .RSTPRE        (rstpre    ),
    .RSTX          (rstx      ),
    .RSTXB         (rstxb     ),
    .RSTY          (rsty      ),
    .RSTZ          (rstz      )
    );
```

6. Detailed function introduction

For detailed usage, please refer to "UG050003_Titan2 Series FPGA Arithmetic Processing Module (APM) User Guide".

(2) GTP_GRS instruction

1. Supported devices

Table 334 GTP_GRS device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

The GTP is used to control the global reset signal.

3. Port description

Table 335 GTP_GRS port description

Port signal	Input / Output	Describe
GRS_N	input	Global reset

4. Instantiation template

```
GTP_GRS GTP_GRS_inst (
.GRS_N(grs_n)
);
```

(3) GTP_START_E1 instruction

1. Supported devices

Table 336 GTP_START_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

This GTP is used to describe the global signal release process for wakeup operations.

3. Port description

Table 337 GTP_START_E1 port description

Port signal	Input / Output	Describe
CLK	Input	Wake up clock
GOE	Input	Global IO output enable
GRS_N	Input	Global reset
GWE	Input	Memory write enable
WAKEUP_OVER	Output	Wake-up end identification signal

4. Instantiation template

```
GTP_START_E1 GTP_START_E1_inst (
.WAKEUP_OVER (wakeup_over),
.CLK         (clk        ),
.GOE         (goe        ),
.GRS_N       (grs_n      ),
.GWE         (gwe        )
);
```

(4) GTP_ADC_E2 instruction

1. Supported devices

Table 338 GTP_ADC_E2 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_ADC_E2 is used to implement the ADC function. Its structural block diagram is shown in the figure below. For detailed functions, please refer to "UG050009_Titan2 Series FPGA Analog-to-Digital Conversion Module (ADC) User Guide".

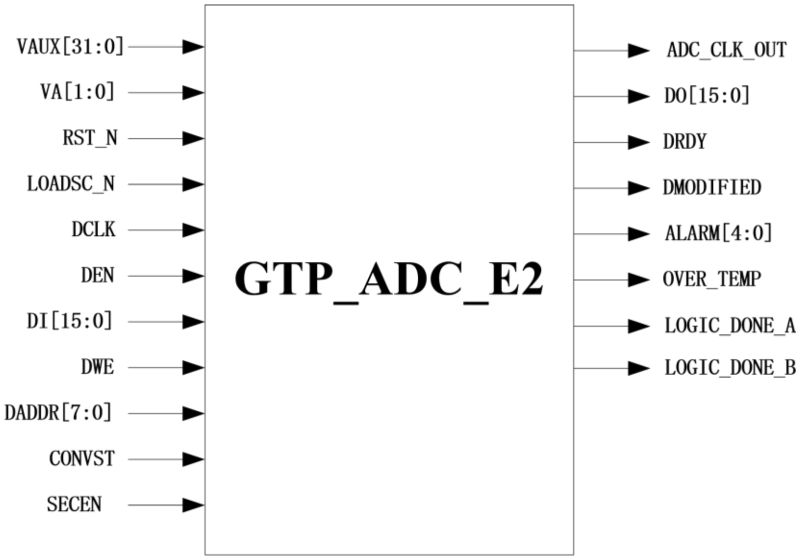


Figure 120 GTP_ADC_E2 structure diagram

3. Port description

Table 339 GTP_ADC_E2 port list

Port Name	Direction	Functional description
VA[1:0]	Input	Dedicated analog input port, VA[1] and VA[0] form a differential pair, VA[0] is the negative terminal, VA[1] is the positive terminal
VAUX[31:0]	Input	Multiplexed analog input port, Multiplex IOB VAUX[30] and VAUX[31] to form a differential pair: VAUX[30] is the negative terminal, VAUX[31] is the positive terminal VAUX[28] and VAUX[29] form a differential pair: VAUX[28] is the negative terminal, VAUX[29] is the positive terminal VAUX[26] and VAUX[27] form a differential pair: VAUX[26] is the negative terminal, VAUX[27] is the positive terminal VAUX[24] and VAUX[25] form a differential pair: VAUX[24] is the negative terminal, VAUX[25] is the positive terminal VAUX[22] and VAUX[23] form a differential pair: VAUX[22] is the negative terminal, VAUX[23] is the positive terminal VAUX[20] and VAUX[21] form a differential pair: VAUX[20] is the negative terminal, VAUX[21] is the positive terminal

Port Name	Direction	Functional description
		VAUX[18] and VAUX[19] form a differential pair: VAUX[18] is the negative terminal, VAUX[19] is the positive terminal VAUX[16] and VAUX[17] form a differential pair: VAUX[16] is the negative terminal, VAUX[17] is the positive terminal VAUX[14] and VAUX[15] form a differential pair: VAUX[14] is the negative terminal, VAUX[15] is the positive terminal VAUX[12] and VAUX[13] form a differential pair: VAUX[12] is the negative terminal, VAUX[13] is the positive terminal VAUX[10] and VAUX[11] form a differential pair: VAUX[10] is the negative terminal, VAUX[11] is the positive terminal VAUX[8] and VAUX[9] form a differential pair: VAUX[8] is the negative terminal, VAUX[9] is the positive terminal VAUX[6] and VAUX[7] form a differential pair: VAUX[6] is the negative terminal, VAUX[7] is the positive terminal VAUX[4] and VAUX[5] form a differential pair: VAUX[4] is the negative terminal, VAUX[5] is the positive terminal VAUX[2] and VAUX[3] form a differential pair: VAUX[2] is the negative terminal, VAUX[3] is the positive terminal VAUX[0] and VAUX[1] form a differential pair: VAUX[0] is the negative terminal, VAUX[1] is the positive terminal
DCLK	Input	APB Clock (1M~100MHz)
DADDR[7:0]	Input	APB operation address bit
DEN	Input	Operation enable, start a read/write operation when high level
SECEN	Input	Transmission signal indicating the second and subsequent cycles
DWE	Input	Write operation enable: 0 for reading, 1 for writing
DI[15:0]	Input	APB data entry
CONVST	Input	Active control signal
RST_N	Input	System reset signal, active low
LOADSC_N	Input	System Control Register Download Static Configuration Value Signal
DO[15:0]	Output	APB data output
DRDY	Output	Jtag operation flag, indicating that Jtag is performing APB operation through JAPB instruction
OVER_TEMP	Output	Over temperature indicator signal
LOGIC_DONE_A	Output	Signal for ADC status register update
LOGIC_DONE_B	Output	Signal for ADC status register update
ADC_CLK_OUT	Output	ADC working clock transmitted to SRB
DMODIFIED	Output	Control register change flag, indicating that after the control register is written by Jtag-APB, the APB operation has not yet been performed
ALARM[4:0]	Output	Alarm indicator signal

4. Parameter description

Table 340 GTP_ADC_E2 parameter list

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
CREG_00H[15:0]	<binary>	16'h0~16'hFFFF	16'h0001	For details, see "UG050009_Titan2 Series FPGA Analog-to-Digital Conversion Module (ADC) User Guide"
CREG_01H[15:0]	<binary>	16'h0~16'hFFFF	16'hC83F	
CREG_02H[15:0]	<binary>	16'h0~16'hFFFF	16'h0009	

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
CREG_31H[13:0]	<binary>	14'h0~14'h3FFF	14'h0	
CREG_03H[15:0]	<binary>	16'h0~16'hFFFF	16'h0	Enable channel: 1 enable; 0 disable
CREG_04H[15:0]	<binary>	16'h0~16'hFFFF	16'h0	Enable channel: 1 enable; 0 disable
CREG_0AH[15:0]	<binary>	16'h0~16'hFFFF	16'h0	Enable channel: 1 enable; 0 disable
CREG_05H[15:0]	<binary>	16'h0~16'hFFFF	16'h0	Select scan mode:1 bipolar; 0 unipolar
CREG_06H[15:0]	<binary>	16'h0~16'hFFFF	16'h0	Select scan mode:1 bipolar; 0 unipolar
CREG_0CH[15:0]	<binary>	16'h0~16'hFFFF	16'h0	Select scan mode:1 bipolar; 0 unipolar
CREG_07H[15:0]	<binary>	16'h0~16'hFFFF	16'h0	Whether to enable averaging:1 enable; 0 disable
CREG_08H[15:0]	<binary>	16'h0~16'hFFFF	16'h0	Whether to enable averaging: 1 enable; 0 disable
CREG_0EH[15:0]	<binary>	16'h0~16'hFFFF	16'h0	Whether to enable averaging:1 enable; 0 disable
CREG_20H[11:0]	<binary>	12'h0~12'hFFF	12'h0	Internal temperature sensor upper threshold setting
CREG_21H[11:0]	<binary>	12'h0~12'hFFF	12'h0	Internal temperature sensor lower threshold setting
CREG_22H[11:0]	<binary>	12'h0~12'hFFF	12'h0	Internal voltage VCC upper threshold setting
CREG_23H[11:0]	<binary>	12'h0~12'hFFF	12'h0	Internal voltage VCC lower threshold setting
CREG_24H[11:0]	<binary>	12'h0~12'hFFF	12'h0	Internal voltage VCCA upper threshold setting
CREG_25H[11:0]	<binary>	12'h0~12'hFFF	12'h0	Internal voltage VCCA lower threshold setting
CREG_26H[11:0]	<binary>	12'h0~12'hFFF	12'h0	Internal voltage VCC_DRM upper threshold setting
CREG_27H[11:0]	<binary>	12'h0~12'hFFF	12'h0	Internal voltage VCC_DRM lower threshold setting
CREG_28H[11:0]	<binary>	12'h0~12'hFFF	12'h0	Internal voltage VCC_CRAM upper threshold setting
CREG_29H[11:0]	<binary>	12'h0~12'hFFF	12'h0	Internal voltage VCC_CRAM lower threshold setting
CREG_2AH[11:0]	<binary>	12'h0~12'hFFF	12'hCC2	Over temperature upper threshold setting
CREG_2BH[11:0]	<binary>	12'h0~12'hFFF	12'hA5B	Over temperature lower threshold setting

5. Instantiation template

```
GTP_ADC_E2 #(
.CREG_00H('b0000000000000001),
.CREG_01H('b1100100000111111),
.CREG_02H('b0000000000001001),
.CREG_31H('b00000000000000),
.CREG_03H('b0000000000000000),
.CREG_04H('b0000000000000000),
.CREG_0AH('b0000000000000000),
.CREG_05H('b0000000000000000),
```

```
.CREG_06H('b0000000000000000),
.CREG_0CH('b0000000000000000),
.CREG_07H('b0000000000000000),
.CREG_08H('b0000000000000000),
.CREG_0EH('b0000000000000000),
.CREG_20H('b000000000000),
.CREG_21H('b000000000000),
.CREG_22H('b000000000000),
.CREG_23H('b000000000000),
.CREG_24H('b000000000000),
.CREG_25H('b000000000000),
.CREG_26H('b000000000000),
.CREG_27H('b000000000000),
.CREG_28H('b000000000000),
.CREG_29H('b000000000000),
.CREG_2AH('b110011000010),
.CREG_2BH('b101001011011)
) GTP_ADC_E2_inst (
.ALARM       (alarm    ),
.DO          (do       ),
.DADDR       (daddr    ),
.DI          (di       ),
.VAUX        (vaux     ),
.ADC_CLK_OUT (adc_clk_out ),
.DMODIFIED   (dmodified),
.DRDY        (drdy     ),
.LOGIC_DONE_A(logic_done_a),
.LOGIC_DONE_B(logic_done_b),
.OVER_TEMP   (over_temp),
.CONVST      (convst   ),
.DCLK        (dclk     ),
.DEN         (den      ),
.DWE         (dwe      ),
.LOADSC_N    (loadsc_n),
.RST_N       (rst_n    ),
.SECEN       (secen    ),
.VA          (va       )
);
```

(5) GTP_RES_CAL_E1 instruction

1. Supported devices

Table 341 GTP_RES_CAL_E1 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_RES_CAL_E1 Impedance calibration for HPIO when describing user mode.

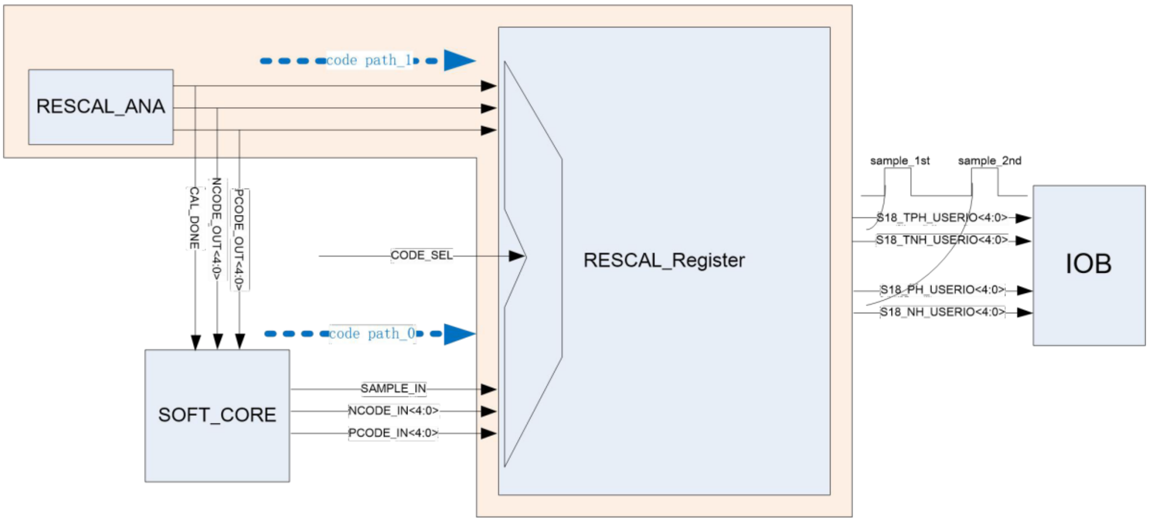


Figure 121 GTP_RES_CAL_E1 main structure

3. Port description

GTP_RES_CAL_E1 All ports are connected to SRB, and the part hard-connected to IOB is not reflected in GTP, and there are no related ports on both sides.

Table 342 GTP_RES_CAL_E1 port list

Signal	I/O	Describe	Initial value	Connect
PCODE_IN<4:0>	I	pcode	11111	SRB
NCODE_IN<4:0>	I	ncode	00000	SRB
SAMPLE_IN	I	code sampling signal	0	SRB
EN	I	rescal enable	1	SRB
CODE_SEL	I	CODE path selection (1:	1	SRB
RST_N	I	internal/0:external)	1	SRB
PCODE_OUT<4:0>	O	rescal reset	11111	SRB
NCODE_OUT<4:0>	O	Calibrated pcode output	00000	SRB
CAL_DONE	O	Calibrate ncode output	0	SRB

4. Parameter description

Table 343 GTP_RES_CAL_E1 parameter list

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
CAL_ENABLE	Boolean	FALSE/TRUE	FALSE	Calibration enable is valid (when CAL_CASCADE is UP/DN, the calibration value source is the upper/lower calibration completion value, the current RESCAL does not need to process the calibration value, you can choose false to reduce static power consumption)
CAL_CASCADE	String	LOCAL/UP/DN	LOCAL	Calibration value source, you can choose current IO or upper and lower IO
CASC_V_ENABLE	Boolean	FALSE/TRUE	FALSE	Enable vertical cascading of DONE signals between BANKs. DCI_MATCH must be set to TRUE when it is valid.

BANK_LOC	String	BANKR5/ BANKR6/ BANKR7	BANKR5	Specify BANK location

5. Instantiation template

```
GTP_RES_CAL_E1 #(
.CAL_ENABLE("FALSE"),
.CAL_CASCADE("LOCAL"),
.CASC_V_ENABLE("FALSE")
) GTP_RES_CAL_E1_inst (
.NCODE_OUT(),// OUTPUT[4:0]
.PCODE_OUT(),// OUTPUT[4:0]
.NCODE_IN(), // INPUT[4:0]
.PCODE_IN(), // INPUT[4:0]
.CAL_DONE(), // OUTPUT
.CODE_SEL(), // INPUT
.EN(),        // INPUT
.RST_N(),     // INPUT
.SAMPLE_IN() // INPUT
);
```

6. Detailed function description

After the chip startup configuration is completed, set EN and RST according to the timing requirements. After the calibration is completed, the code will be transmitted to the IOB.

(6) GTP_PCIEGEN3 instruction

1. Supported devices

Table 344 GTP_PCIEGEN3 device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_PCIEGEN3 (Peripheral Component Interconnect Express) high-speed serial interface module is compatible with PCIE3.0 protocol. Main features include: support Gen1 and Gen2, Gen3 three rates (2.5Gb/s, 5Gb/s or 8Gb/s); support EP and RC Two modes and support x1, x2, x4 and x8 at the same time; support 2 Physical functions, and up to 6 Virtual functions; TLP package max_payload can be flexibly configured between 128Bytes, 256 Bytes, 512 Bytes, and 1024 Bytes; BAR registers can be configured ;Support Legacy, MSI and MSIX three interrupt mechanisms at the same time; Power Management, Lane Reversal, Lane Polarity Inversion can all be realized. The brief block diagram of the structure is as follows:

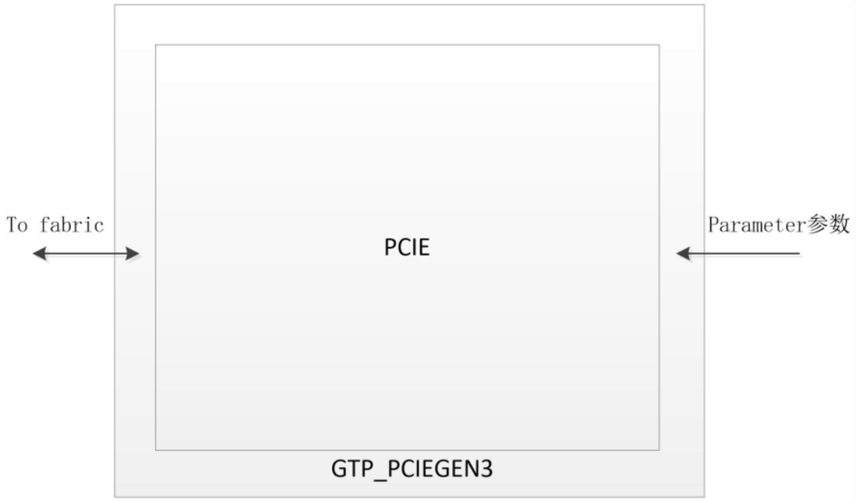


Figure 122 GTP_PCIEGEN3 structure diagram

3. Port description

Table 345 GTP_PCIEGEN3 port list

Port Name	Direction	Bit width	Functional description
PCLK	I	1	PIPE clock from external PHY
PCLK_DIV2	I	1	pclk_div2 clock from external PHY
MEM_CLK	I	1	mem interface clock, PIPE clock from external PHY
USER_CLK	I	1	User clk, the frequency is fixed after configuration
BUTTON_RST	I	1	Button reset, Active high
POWER_UP_RST	I	1	Warm reset, Active high
PERST	I	1	Reset from Socket, Active high
USER_RST_N	O	1	User side reset
PHY_RST_N	O	1	For phy rst,Active low
AXIS_MASTER1_TVALID	O	1	Active high to indicate that the ctrl is driving a valid transfer
AXIS_MASTER1_TDATA	O	256	Primary payload provide to user logic
AXIS_MASTER1_TKEEP	O	8	Ativie high to enable related DW in axis_master_tdata; axis_master_tkeep[7] control axis_master_tdata[255:224]; axis_master_tkeep[6] control axis_master_tdata[223:192]; axis_master_tkeep[5] control axis_master_tdata[191:160]; axis_master_tkeep[4] control axis_master_tdata[159:128]; axis_master_tkeep[3] control axis_master_tdata[127:96]; axis_master_tkeep[2] control axis_master_tdata[95:64]; axis_master_tkeep[1] control axis_master_tdata[63:32]; axis_master_tkeep[0] control axis_master_tdata[31:0];
AXIS_MASTER1_TUSER	O	42	[7:0] eop: Position of received tlp end: eop[0] = 1 means tlp end is located at axis_master1_tdata[31:0]; eop[1] = 1 means tlp end is located at axis_master1_tdata[63:32]; eop [7] = 1 means that tlp end is located in axis_master1_tdata[255:224]. eop[7:0],there may be at most 2 bits valid at the same time.

Port Name	Direction	Bit width	Functional description
			[15:8] sop: Position of received tlp start: sop[0] = 1 means tlp start is located at axis_master1_tdata[31:0]; sop[1] = 1 means tlp start is located at axis_master1_tdata[63:32]; ... sop[7] = 1 means that tlp is located in axis_master1_tdata[255:224]. sop[7:0], and at most 2 bits may be valid at the same time. [18:16]: radm_trgt1_vfunc_num in first tlp 19 : radm_trgt1_vfunc_active in first tlp 20 : radm_trgt1_func_num in first tlp 21: radm_trgt1_tlp_abort (Indicates to your application to drop the TLP because of malformed TLP on TRGT1, ECRC error, or completion lookup failures) in first tlp 22: radm_trgt1_dllp_abort (Indicates to your application to drop the TLP on TRGT1 because of a Data Link Layer error such as LCRC or otherwise) in first tlp 23: radm_trgt1_ecrc_err (Indicates to your application to drop the TLP because of an ECRC error in the received TLP on TRGT1) in first tlp 24 : radm_trgt1_cpl_last [33:25]: second tlp information, the specific meaning is consistent with [24:16].
AXIS_MASTER0_TVALID	O	1	Active high to indicate that the ctrl is driving a valid transfer
AXIS_MASTER0_TREADY	I	1	Active high to indicate that the user can accept a transfer in the current cycle
AXIS_MASTER0_TDATA	O	256	Primary payload provide to user logic
AXIS_MASTER0_TKEEP	O	8	Ativie high to enable related DW in axis_master_tdata; axis_master_tkeep[7] control axis_master_tdata[255:224]; axis_master_tkeep[6] control axis_master_tdata[223:192]; axis_master_tkeep[5] control axis_master_tdata[191:160]; axis_master_tkeep[4] control axis_master_tdata[159:128]; axis_master_tkeep[3] control axis_master_tdata[127:96]; axis_master_tkeep[2] control axis_master_tdata[95:64]; axis_master_tkeep[1] control axis_master_tdata[63:32]; axis_master_tkeep[0] control axis_master_tdata[31:0];
AXIS_MASTER0_TLAST	O	1	Active high to indicate last valid transfer of a packet from ctrl, valid when axis_master0_tvalid is high
AXIS_MASTER0_TUSER	O	13	Sideband information transmitted alongside axis_master_tdata: [2:0]: radm_trgt1_vfunc_num 3: radm_trgt1_vfunc_active 4: radm_trgt1_func_num 5: radm_trgt1_tlp_abort (Indicates to your application to drop the TLP because of malformed TLP on TRGT1, ECRC error, or completion lookup failures) 6: radm_trgt1_dllp_abort (Indicates to your application to drop the TLP on TRGT1 because of a Data Link Layer error such as LCRC or otherwise) 7: radm_trgt1_ecrc_err (Indicates to your application to drop the TLP because of an ECRC error in the received TLP on TRGT1) 8:radm_trgt1_cpl_last: Indicates the last completion TLP of a split

Port name	Direction	Bit width	Functional description
			completion transaction. [11:9]:radm_trgt1_in_membar_range[2:0](Indicates which ofthe configured BARs contains the target address in the received TLP) 12: radm_trgt1_rom_in_range (Indicates that the target address in the received TLP in range of the expansion ROM)
USER_RCVD_NP_READY	I	1	Every time the user pulls up user_rcvd_np_ready, it means that an np request output by the core can be received. The user side uses this signal to prevent the p request from being blocked due to the inability to process the np request output by the core. When the user does not need to perform np type data management, the signal can be tied to 1
CORE_AVL_NP_CNT	O	6	Indicates the number of np requests left by the current core that can be output. Due to internal delays, this signal will not update the real number of np that can be output inside the core in real time, and there will be a delay of several clks.
USER_RCVD_P_READY	I	1	Every time the user pulls up user_rcvd_np_ready, it means that a p request output by the core can be received. When the user does not need to perform p-type data management, the signal can be tied to 1
CORE_AVL_P_CNT	O	6	Indicates the number of p requests that the current core can output. Due to the internal delay, this signal will not update the real p number that can be output inside the core in real time, and there will be a delay of several clk.
AXIS_SLAVE0/1/2_TREADY	O	1	Active high to indicate that the ctrl can accept a transfer in the current cycle
AXIS_SLAVE0/1/2_TVALID	I	1	Active high to indicate that the user is driving a valid transfer
AXIS_SLAVE0/1/2_TDATA	I	256	Primary payload provide to ctrl
AXIS_SLAVE0/1/2_TLAST	I	1	Active high to indicate last valid transfer of a packet to ctrl
AXIS_SLAVE0/1/2_TUSER	I	6	5： func number [4:2]: vfunc number 1： vfunc active 0： bad eot
RADM_TIMEOUT_FUNC_NUM	O	2	The function Number of the timed out completion. Function numbering starts at '0'. bit 1 is controlled by radm_timeout[1] whether it is valid or not, bit[0] is controlled by radm_timeout[0] whether it is valid
RADM_TIMEOUT_VFUNC_NUM	O	6	Indicates which virtual function (VF) had a completion timeout Bit 5:3 is controlled by radm_timeout[1] whether it is valid, Bit 2:0 is controlled by radm_timeout[0] whether it is valid or not
RADM_TIMEOUT_VFUNC_ACTIVE	O	2	Indicates that a virtual function (VF) had a completion timeout. bit 1 is controlled by radm_timeout[1] whether it is valid, bit[0] is controlled by radm_timeout[0] whether it is valid
RADM_CPL_TIMEOUT	O	2	Indicates that the completion TLP for a request has not been received within the expected time window, there may be two comletions with timeout at the same time, so radm_timeout has 2 bits, bit 0 corresponds to radm_timeout_func_num[0] radm_timeout_vfunc_num[2:0], radm_timeout_vfunc_active[0], radm_timo ut_cpl_tc[2:0], radm_timeout_cpl_tag[7:0], radm_timeout_cpl_attr[1:0], radm_timeout_cpl_len[10:0], bit 1 corresponds to other bits
RADM_TIMEOUT_CPL_TC	O	6	The Traffic Class of the timed out completion Bit 5:3 is controlled by radm_timeout[1], and Bit 2:0 is controlled by radm_timeout[0].
RADM_TIMEOUT_CPL_TAG	O	16	The Tag field of the timed out completion Bit 15:8 is controlled by radm_timeout[1] whether it is valid, Bit 7:0 is controlled by radm_timeout[0] whether it is valid

Port Name	Direction	Bit width	Functional description
RADM_TIMEOUT_CPL_ATTR	O	4	The Attributes field of the timed out completion. Bit 3:2 is controlled by radm_timeout[1], and Bit 1:0 is controlled by radm_timeout[0].
RADM_TIMEOUT_CPL_LEN	O	22	Length (in bytes) of the timed out completion. For a split completion, it indicates the number of bytes remaining to be delivered when the completion timed out. Bit 21:11 is controlled by radm_timeout[1]. Bit 10:0 is controlled by radm_timeout[0] whether it is valid or not
MSG_RCVD_TYPE	O	5	Received message type. Controlled by msg_rcvd, see pcie3_hard_msg_rcv_adapter description for specific msg_rcvd_type decoding
MSG_RCVD_DATA	O	8	Received message date. Controlled by msg_rcvd, see pcie3_hard_msg_rcv_adapter description for specific msg_rcvd_type decoding
MSG_RCVD	O	1	Message received valid
APP_UNLOCK_MSG	I	1	Request from your application to generate an Unlock message. You must assert this signal for one clock cycle. The controller does not return an acknowledgment or grant signal. You must not pulse the same signal again, until the previous message has been transmitted
APPS_PM_XMT_TURNOFF	I	1	Request from your application to generate a PM_Turn_Off message. You must assert this signal for one clock cycle. The controller does not return an acknowledgment or grant signal. You must not pulse the same signal again, until the previous message has been transmitted
APP_OBFF_IDLE_MSG_REQ	I	1	Request from your application to generate an 'IDLE' OBFF message. Only usable in a downstream port
APP_OBFF_OBFF_MSG_REQ	I	1	Request from your application to generate an 'OBFF' OBFF message. Only usable in a downstream port.
APP_OBFF_CPU_ACTIVE_MSG_REQ	I	1	Request from your application to generate a 'CPU Active' OBFF message. Only usable in a downstream port
APP_OBFF_MSG_GRANT	I	1	Indicates that the controller has accepted your request to generate an OBFF message. Only usable in a downstream port
CFG_LTR_M_EN	O	1	The LTR Mechanism Enable field of the Device Control 2 register of function 0.
APP_LTR_MSG_REQ	I	1	Indicates that your application is requesting to send an LTR message. Once asserted, app_ ltr_msg_req must remain asserted until the controller asserts app_ltr_msg_grant
APP_LTR_MSG_GRANT	O	1	Indicates that the controller has accepted your request to send an LTR message.
APP_LTR_MSG_LATENCY	I	32	LTR message that your application is requesting to send
APP_LTR_MSG_FUNC_NUM	I	1	Function number in your application that is requesting to send an LTR message. Function numbering starts at '0'.
CFG_DISABLE_LTR_CLR_MSG	O	1	Disable the autonomous generation of LTR clear message
DBI_ADDR	I	32	Address of the configuration register for the current DBI access [31:2] register address, must be dword-aligned [1] not used [0] DBI access direction, 0: internal register 1: external register via ELBI

Port Name	Direction	Bit width	Functional description
DBI_DIN	I	32	Write data bus to the selected configuration register
DBI_CS	I	1	Chip select input to access the CDM or ELBI
DBI_CS2	I	1	Additional chip select that enables writing to BAR mask registers
DBI_WR	I	4	Indicates the configuration register access type (read or write).
APP_DBI_RO_WR_DISABLE	I	1	DBI Read-only Write Disable 0:MISC_CONTROL_1_OFF.DBI_RO_WR_EN register field is read-write. 1: MISC_CONTROL_1_OFF.DBI_RO_WR_EN register field is forced to 0 and is read-only.
LBC_DBI_ACK	O	1	Indicates that the requested read or write operation to the selected configuration register is complete.
LBC_DBI_DOUT	O	32	Read data bus from the selected configuration register.
DBI_VFUNC_NUM	I	3	Indicates which VF is being accessed through the DBI
DBI_VFUNC_ACTIVE	I	1	Indicates that a VF is being accessed through the DBI
DBI_FUNC_NUM	I	1	The (physical) function number of the current DBI access
SEDI	I	2	Serial signal input, driven by slave, bit 0 for func 0, bit 1 for func 1
SEDI_ACK	I	2	Serial signal input ACK, driven by slave bit 0 for func 0, bit 1 for func 1
SEDO	O	2	Serial signal output, driven by master bit 0 for func 0, bit 1 for func 1
SEDO_EN	O	2	Serial signal output enable, driven by master bit 0 for func 0, bit 1 for func 1
VEN_MSI_REQ	I	1	Request from your application to send an MSI when MSI is enabled. When MSI-X is enabled instead of MSI, assertion of ven_msi_req causes the controller to generate an MSI-X message. Once asserted, ven_msi_req must remain asserted until the controller asserts ven_msi_grant.
VEN_MSI_FUNC_NUM	I	1	The function number of the MSI request. Function numbering starts at '0'.
VEN_MSI_VFUNC_NUM	I	3	Identifies the VF which is making the MSI request, To ease timing during synthesis, the PCIe controller starts numbering VFs at 0.
VEN_MSI_VFUNC_ACTIVE	I	1	Indicates that the MSI request is coming from a VF
VEN_MSI_TC	I	3	Traffic Class of the MSI request, valid when ven_msi_req is asserted.
VEN_MSI_VECTOR	I	5	Used to modulate the lower five bits of the MSI Data register when multiple message mode is enabled.
VEN_MSI_GRANT	O	1	One-cycle pulse that indicates that the controller has accepted the request to send an MSI
CFG_MSI_MASK_UPDATE	O	1	Indication msi mask bit updated in pf or vf
CFG_MSI_PENDING	I	64	Indication from application about which functions have a pending associated message.
CFG_VF_MSI_PENDING	I	192	Contents of the Per Vector Pending register in the VF MSI Capability structure.
MSIX_ADDR	I	64	The address value for the MSI-X
MSIX_DATA	I	32	The data value for the MSI-X
CFG_MSIX_FUNC_MASK	O	2	The function Mask bit of the MSI-X Control register in the MSI-X Capability structure.
CFG_VF_MSIX_FUNC_MASK	O	6	MSIX_EN bits from the Message Control Register in the MSI-X Capability register of each VF.

Port Name	Direction	Bit width	Functional description
CFG_VF_MULTI_MSI_EN	O	18	The Multiple Message Capable field of the MSI Control register in the VF MSI Capability structure
CFG_VF_MSIX_EN	O	6	MSIX_EN bits from the Message Control Register in the MSI-X Capability register of each VF.
CFG_VF_MSI_EN	O	6	Indicates when the VF MSI Capability structure is enabled for each configured function
CFG_MULTI_MSI_EN	O	6	The Multiple Message Enabled field of the MSI Control register in the MSI Capability structure
CFG_MSIX_EN	O	2	The MSI-X Enable bit of the MSI-X Control register in the MSI-X Capability structure.
CFG_MSI_EN	O	2	Indicates that MSI is enabled (INTx message is not sent), one bit per configured function.
SYS_INT	I	2	When sys_int goes from low to high, the controller generates an Assert_INTx Message. When sys_int goes from high to low, the controller generates a Deassert_INTx Message. 2 bit sys_int for INTa and INTb respectively. Every time the User side pulls up or pulls down sys_int, it needs to wait for the int_grt response before the next pull up or down. 2bit sys_int shares 1bit grant.
INT_GRT	O	1	Low to high : assert_intx message has been sent High to low :deassert_intx_message has been sent
PM_XTLH_BLOCK_TLP	O	1	Indicates that your application must stop generating new outgoing request TLPs due to the current power management state
TRAINING_RST_N	O	1	Hot reset from upstream component
PM_DSTATE	O	6	The current power management D-state of the function: 000b: D0 001b: D1 010b: D2 011b: D3 100b: Uninitialized Other values: Not applicable
PM_VF_DSTATE	O	18	The current power management D-state of the virtual function: 000b: D0 001b: D1 010b: D2 011b: D3 100b: Uninitialized Other values: Not applicable
PM_VF_PME_EN	O	6	Enable bit in the VF PMCSR
PM_VF_STATUS	O	6	PME Status bit from the VF PMCSR
AUX_PM_EN	O	2	Auxiliary Power Enable bit in the Device Control register
PM_PME_EN	O	2	PME Enable bit in the PMCSR
PM_STATUS	O	2	PME Status bit from the PMCSR
WAKE	O	1	Wake Up. Wake up from power management unit.
PM_MASTER_STATE	O	5	Power management master FSM state.
PM_SLAVE_STATE	O	5	Power management slave FSM state.
RDLH_LINK_UP	O	1	DL layer link up

Port Name	Direction	Bit width	Functional description
SMLH_LTSSM_STATE	O	6	Current state of the LTSSM. Encoding is defined as follows: 6'h00: S_DETECT_QUIET 6'h01: S_DETECT_ACT 6'h02: S_POLL_ACTIVE 6'h03: S_POLL_COMPLIANCE 6'h04: S_POLL_CONFIG 6'h05: S_PRE_DETECT_QUIET 6'h06: S_DETECT_WAIT 6'h07: S_CFG_LINKWD_START 6'h08: S_CFG_LINKWD_ACEPT 6'h09: S_CFG_LANENUM_WAI 6'h0A: S_CFG_LANENUM_ACEPT 6'h0B: S_CFG_COMPLETE 6'h0C: S_CFG_IDLE 6'h0D: S_RCVRY_LOCK 6'h0E: S_RCVRY_SPEED 6'h0F: S_RCVRY_RCVRCFG 6'h10: S_RCVRY_IDLE 6'h11: S_L0 6'h12: S_L0S 6'h13: S_L123_SEND_EIDLE 6'h14: S_L1_IDLE 6'h15: S_L2_IDLE 6'h16: S_L2_WAKE 6'h17: S_DISABLED_ENTRY 6'h18: S_DISABLED_IDLE 6'h19: S_DISABLED 6'h1A: S_LPBK_ENTRY 6'h1B: S_LPBK_ACTIVE 6'h1C: S_LPBK_EXIT 6'h1D: S_LPBK_EXIT_TIMEOUT 6'h1E: S_HOT_RESET_ENTRY 6'h1F: S_HOT_RESET 6'h20: S_RCVRY_EQ0 6'h21: S_RCVRY_EQ1 6'h22: S_RCVRY_EQ2 6'h23: S_RCVRY_EQ3
SMLH_LINK_UP	O	1	Mac layer link up
SMLH_REQ_RST	O	1	Early version of the link_req_rst_not signal,active hign
LINK_REQ_RST	O	1	Reset request because the link has gone down or the controller received a hot-reset request active high
CFG_VF_BME	O	6	Bus master enable bit from the Control Register in the PCI header of each VF
CFG_VF_EN	O	2	Identifies those physical functions that have virtual functions enabled
CFG_NUM_VF	O	6	The number of virtual functions in each physical function.
CFG_START_VFI	O	6	The index of the first VF in every PF
CFG_TPH_REQ_EN	O	2	The TPH Requester Enabled field of each TPH Requester Control register.

Port Name	Direction	Bit width	Functional description
CFG_PF_TPH_ST_MODE	O	6	Steering Tag Mode of Operation for Physical Function
CFG_VF_TPH_ST_MODE	O	18	Steering Tag Mode of Operation for Virtual Function
CFG_VF_TPH_REQ_EN	O	6	The TPH Requester Enabled field of each VF TPH Requester Control register.
DPA_SUBSTATE_UPDATE	O	2	The controller asserts it for one core_clk cycle when the Substate Control field of the DPA Control Register has been updated with a new value for the associated function.
CFG_ATOMIC_REQ_EN	O	2	The AtomicOp Requester Enable field (PCIE_CAP_ATOMIC_REQ_EN) of the Device Control 2 register.
CFG_ATOMIC_EGRESS_BLOCK	O	2	The AtomicOp Egress Blocking field (PCIE_CAP_ATOMIC_EGRESS_BLK) of the Device Control 2 register
CFG_ARI_FWD_EN	O	2	ARI Forwarding Enabled (DSP)
CFG_SEND_COR_ERR	O	2	Sent Correctable Error. Controller has sent a message towards the Root Complex
CFG_SEND_NF_ERR	O	2	Sent non-fatal Error. Controller has sent a message towards the Root Complex
CFG_SEND_F_ERR	O	2	Sent fatal Error. Controller has sent a message towards the Root Complex
CFG_INT_DISABLE	O	2	When high a functions ability to generate INTx messages is Disabled
CFG_NO_SNOOP_EN	O	2	Contents of the "Enable No Snoop" field in the "Device Control and Status" register
CFG_RELAX_ORDER_EN	O	2	Contents of the "Enable Relaxed Ordering" field in the "Device Control and Status
CFG_LINK_AUTO_BW_INT	O	1	The controller asserts cfg_link_auto_bw_int when all of the following conditions are true: The INTx assertion disable bit in the Command register is 0, and The Link Autonomous Bandwidth Interrupt Enable bit in the Link Control register is set to 1, and The Link Autonomous Bandwidth Interrupt Status bit in the Link Status register is set to 1.
CFG_LINK_AUTO_BW_MSI	O	1	The controller sets this pin when following conditions are true: MSI or MSI-X is enabled. The Link Autonomous Bandwidth Status register (Link Status register bit 15) is updated. The Link Autonomous Bandwidth Interrupt Enable (Link Control register bit 11) is set
CFG_BW_MGT_INT	O	1	The controller asserts cfg_bw_mgt_int when all of the following conditions are true: The INTx Assertion Disable bit in the Command register is 0, and The Bandwidth Management Interrupt Enable bit in the Link Control register is set to 1, and The Bandwidth Management Interrupt Status bit in the Link Status register is set to 1.
CFG_BW_MGT_MSI	O	1	The controller sets this pin when following conditions are true: MSI or MSI-X is enabled. The Link Bandwidth Management Status register (Link Control Status register bit 14) is updated The Link Bandwidth Management Interrupt Enable (Link Control register bit 10) is set.

Port Name	Direction	Bit width	Functional description
CFG_LINK_EQ_REQ_INT	O	1	Interrupt indicating to your application that the Link Equalization Request bit in the Link Status 2 Register has been set and the Link Equalization Request Interrupt Enable (Link Control 3 Register bit 1) is set.
RBAR_CTRL_UPDATE	O	2	Indicates that a resizable BAR control register has been updated: 1 bit per Physical function
CFG_IDO_REQ_EN	O	2	ID-Based Ordering Requests Enabled
CFG_IDO_CPL_EN	O	2	ID-Based Ordering Completions Enabled
CFG_SYS_ERR_RC	O	2	System error detected. A one-clock-cycle pulse that indicates if any device in the hierarchy reports any of the following errors and the associated enable bit is set in the Root Control register: ERR_COR, ERR_FATAL, ERR_NONFATAL. Also asserted when an internal error is detected. There is one bit of cfg_sys_err_rc assigned to each configured function.
CFG_AER_RC_ERR_INT	O	2	Asserted when a reported error condition causes a bit to be set in the Root Error Status register and the associated error message reporting enable bit is set in the Root Error Command register. cfg_aer_rc_err_int is set when the RC internally generates an error or when an error message is received by the RC. Because the RC itself generates it, this needs to be propagated up to the system software which would then need to read the error registers to see which error occurred. *Note:*This signal is used when MSI/MSI-X is NOT enabled; otherwise see cfg_aer_rc_err_msi.
CFG_AER_RC_ERR_MSI	O	2	The controller asserts cfg_aer_rc_err_msi for one clock cycle when all of the following conditions are true: • MSI or MSI-X is enabled. • reported error condition causes a bit to be set in the Root Error Status register. • the associated error message reporting enable bit is set in the Root Error Command register. The controller does not check if the associated MSI vector (asserted cfg_aer_int_msg_num) is unmasked. It is up to the application to check whether the vector is masked or unmasked
CFG_PME_INT	O	2	The controller asserts cfg_pme_int when all of the following conditions are true: • The INTx Assertion Disable bit in the Command register is 0. • The PME Interrupt Enable bit in the Root Control register is set to 1. • The PME Status bit in the Root Status register is set to 1. The cfg_pme_msi output is a pulse signal (only asserted for one clock cycle). But cfg_pme_int is a level signal; essentially an AND of the PME interrupt enable and receipt of the pm_pme message.
CFG_PME_MSI	O	2	The controller asserts cfg_pme_msi (as a one-cycle pulse) when all of the following conditions are true: • MSI or MSI-X is enabled. • The PME Interrupt Enable bit in the Root Control register is set to 1. • The PME Status bit in the Root Status register is set to 1. The controller does not check if the associated MSI vector (asserted cfg_pcie_cap_int_msg_num) is unmasked. It is up to the application to check whether the vector is masked or unmasked

Port Name	Direction	Bit width	Functional description
CFG_CRS_SW_VIS_EN	O	2	Indicates the value of the CRS Software Visibility enable bit in the Root Control register. Applicable only for RC devices.
RADM_Q_NOT_EMPTY	O	1	Level indicating that the receive queues contain TLP header/data
RADM_QOVERFLOW	O	1	Pulse indicating that one or more of the P/NP/CPL receive queues have overflowed
CFG_BUS_MASTER_EN	O	2	The state of the bus master enable bit in the PCI-compatible Command register.
CFG_2ND_RESET	O	1	Secondary Bus Reset. Indicates that your application has requested this downstream port to start Link hot reset. This signal is asserted when your application writes to the Secondary Bus Reset field (bit 6) of the Bridge Control Register (Offset 0x3E) in the Type 1 Configuration Space header. Setting this bit triggers a hot reset the corresponding PCI Express port.
CFG_MAX_PAYLOAD_SIZE	O	6	The value of the Max_Payload_Size field in the Device Control register. There are 3 bits of cfg_max_payload_size assigned to each configured function.
CFG_RCB	O	2	The value of the RCB bit in the Link Control register. There is 1 bit of cfg_rcb assigned to each configured function.
CFG_TC_ENABLE	O	8	Indicates the enabled TCs. TCn is enabled when bit[n] is 1.
CFG_MEM_SPACE_EN	O	2	The state of the Memory Space Enable bit in the PCI-compatible Command register. There is 1 bit of cfg_mem_space_en assigned to each configured function
CFG_MAX_RD_REQ_SIZE	O	6	The value of the Max_Read_Request_Size field in the Device Control register. There are 3 bits of cfg_max_rd_req_size assigned to each configured function
CFG_EXT_TAG_EN	O	2	When enabled, controller supports up to 8-bit tag values.
CFG_REG_SERREN	O	2	PF's SERR# Enable registers value in Command Register of Type0 Header, for setting error status registers of external VFs
CFG_COR_ERR_RPT_EN	O	2	PF's Correctable Error Reporting Enable registers value in Device Control Register of PCIe Capability
CFG_NF_ERR_RPT_EN	O	2	PF's Non-Fatal Error Reporting Enable registers value in Device Control Register of PCIe Capability
CFG_F_ERR_RPT_EN	O	2	PF's Fatal Error Reporting Enable registers value in Device Control Register of PCIe Capability
CFG_PBUS_NUM	O	8	The primary bus number assigned to the function
CFG_PBUS_DEV_NUM	O	5	The device number assigned to the function
CFG_OBFF_EN	O	2	The OBFF Enable field of the Device Control 2 register of function 0.
CFG_PWR_BUDGET_VALID	I	1	cfg_pwr_budget_data_reg valid. When the input cfg_pwr_budget_data_reg changes, pull the signal high for 1 cycle
CFG_PWR_BUDGET_SEL	O	2	One cycle pulse signal indicates new contents in Data Select Register. Something changed the value of Data Select Register
CFG_PWR_BUDGET_FUNC_NUM	I	1	Function # of data register above (because this capability is per function). Function numbering starts at '0'.
CFG_PWR_BUDGET_DATA_SEL_REG	O	8	Contents of the Data Select Register

Port Name	Direction	Bit width	Functional description
CFG_PWR_BUDGET_DATA_REG	I	32	New Data Register value. Data needs to be held
TX_LANE_FLIP_EN	I	1	Performs manual lane reversal for transmit lanes
TPH_RD_DATA_VALID	I	1	Read data valid
RX_LANE_FLIP_EN	I	1	Performs manual lane reversal for receive lanes
APP_XFER_PENDING	I	1	Indicates that your application has transfers pending and prevents the controller from entering L1.
APP_VF_REQ_RETRY_EN	I	6	Provides a per Virtual Function (VF) capability to defer incoming configuration requests until initialization is complete
APP_REQ_RETRY_EN	I	1	Provides a capability to defer incoming configuration requests until initialization is complete
APP_REQ_EXIT_L1	I	1	Application request to Exit L1
APP_REQ_ENTR_L1	I	1	Application request to Enter L1 ASPM state
APP_READY_ENTR_L23	I	1	Application Ready to Enter L23.
APP_PF_REQ_RETRY_EN	I	2	Provides a per Physical Function (PF) capability to defer incoming configuration requests until initialization is complete
APP_LTSSM_ENABLE	I	1	Driven low by your application after cold, warm or hot reset to hold the LTSSM in the Detect state until your application is ready for the link training to begin
APP_CLK_PM_EN	I	1	Clock PM feature enabled by application
DEVICE_TYPE	I	4	Device/port type. Indicates the specific type of this PCI Express function. 3'b000: PCI Express endpoint 3'b001: Legacy PCI Express endpoint 3'b100: Root port of PCI Express root complex
APP_INIT_RST	I	1	Request from your application to send a hot reset to the upstream port. The hot reset request is sent when a single cycle pulse is applied to this pin. In an upstream port, you should set this input to '0'.
APPS_PM_XMT_PME	I	2	Wake Up. If PME is enabled and PME support is configured for current PMCSR D-state asserting this signal causes the controller to wake from either L1 or L2 state. When the controller has transitioned back to the L0 state it transmits a PME message and set the PME_Status. Upon receiving the PME message the root complex should clear the PME_Status and change the D-state back to D0.
APPS_PM_VF_XMT_PME	I	6	Wake Up command to VF. Used by application logic to wake up the PMC state machine from the D3 state. Upon wake-up, the controller sends a PM_PME Message. Needs to be asserted for one clock cycle. Each bit field corresponds to one of the NVF virtual functions. Your application knows how many VFs are assigned to each PF, and by using this knowledge, it can identify the parent PF and the VF number within that PF.
APP_HDR_VALID	I	1	One-clock-cycle pulse indicating that the data app_hdr_log,
APP_HDR_LOG	I	128	The header of the TLP that contained the error indicated app_err_bus, valid when app_hdr_valid is asserted; Provide app_tlp_prfx_log before app_hdr_valid is asserted if needed
APP_ERR_BUS	I	13	The type of error that your application detected
APP_ERR_ADVISORY	I	1	Indicates that your application error is an advisory error.
APP_ERR_FUNC_NUM	I	1	The number of the function that is reporting the error indicated app_err_bus, valid when app_hdr_valid is asserted

Port Name	Direction	Bit width	Functional description
APP_ERR_VFUNC_NUM	I	3	The number of the virtual function that is reporting the error indicated app_err_bus, valid when app_hdr_valid is asserted
APP_ERR_VFUNC_ACTIVE	I	1	Indicates the function number in app_err_vfunc_num is valid
DYN_DEBUG_INFO_SEL	I	6	Dynamic debug_info_mux selection
APP_RAS_DES_SD_HOLD_LTSSM	I	1	Hold and release LTSSM. For as long as this signal is '1', thecontroller stays in the current LTSSM.
DIAG_CTRL_BUS	I	3	Diagnostic Control Bus, 01: Insert LCRC error by inverting the LSB of LCRC 10: Insert ECRC error by inverting the LSB of ECRC 1xx: Select Fast Link Mode
DEBUG_INFO_MUX	O	143	When sc_dyn_debug_sel_en =0, use sc_debug_info_sel sc_dyn_debug_sel_en =1, use dyn_debug_info_sel *_debug_info_sel=0 [78:0] cdm_ras_des_sd_info_common[78:0] [81:79] pm_curnt_state[2:0] [82] pm_l1_entry_started [88:83] diag_status_bus[520:515] *_debug_info_sel=1 [77:0] cdm_ras_des_sd_info_l0 *_debug_info_sel=2 [77:0] cdm_ras_des_sd_info_l1 *_debug_info_sel=3 [77:0] cdm_ras_des_sd_info_l2 *_debug_info_sel=4 [77:0] cdm_ras_des_sd_info_l3 *_debug_info_sel=5 [77:0] cdm_ras_des_sd_info_l4 *_debug_info_sel=6 [77:0] cdm_ras_des_sd_info_l5 *_debug_info_sel=7 [77:0] cdm_ras_des_sd_info_l6 *_debug_info_sel=8 [77:0] cdm_ras_des_sd_info_l7 *_debug_info_sel=9 0 cfg_bad_dllp_err_sts 1 cfg_bad_tlp_err_sts 2 cfg_corrected_internal_err_sts

Port Name	Direction	Bit width	Functional description
			3 cfg_dl_protocol_err_sts 4 cfg_ecrc_err_sts 5 cfg_mlf_tlp_err_sts 6 cfg_rcvr_err_sts 7 cfg_rcvr_overflow_err_sts 8 cfg_replay_number_rollover_err_sts 9 cfg_replay_timer_timeout_err_sts 10 cfg_surprise_down_er_sts 11 cfg_uncor_internal_err_sts [20:12] diag_status_bus[529:521] [23:21] diag_status_bus[534:532] [51:24] diag_status_bus[858:831] [54:52] diag_status_bus[933:931] 55 cfg_fc_protocol_err_sts *_debug_info_sel=10 0 cfg_msi_64[0] [64:1] cfg_msi_addr[63:0] [96:65] cfg_msi_data[31:0] [97] cfg_msi_ext_data_en[0] *_debug_info_sel=11 0 cfg_msi_64[1] [64:1] cfg_msi_addr[127:64] [96:65] cfg_msi_data[63:32] [97] cfg_msi_ext_data_en[1] *_debug_info_sel=12 0 cfg_vf_msi_64[0] [64:1] cfg_vf_msi_addr[63:0] [96:65] cfg_vf_msi_data[31:0] [97] cfg_vf_msi_ext_data_en[0] *_debug_info_sel=13 0 cfg_vf_msi_64[1] [64:1] cfg_vf_msi_addr[127:64] [96:65] cfg_vf_msi_data[63:32] [97] cfg_vf_msi_ext_data_en[1] *_debug_info_sel=14 0 cfg_vf_msi_64[2] [64:1] cfg_vf_msi_addr[191:128] [96:65] cfg_vf_msi_data[95:64] [97] cfg_vf_msi_ext_data_en[2] *_debug_info_sel=15 0 cfg_vf_msi_64[3]

Port Name	Direction	Bit width	Functional description
			[64:1] cfg_vf_msi_addr[255:192] [96:65] cfg_vf_msi_data[127:96] [97] cfg_vf_msi_ext_data_en[3] *_debug_info_sel=16 0 cfg_vf_msi_64[4] [64:1] cfg_vf_msi_addr[319:256] [96:65] cfg_vf_msi_data[159:128] [97] cfg_vf_msi_ext_data_en[4] *_debug_info_sel=17 0 cfg_vf_msi_64[5] [64:1] cfg_vf_msi_addr[383:320] [96:65] cfg_vf_msi_data[191:160] [97] cfg_vf_msi_ext_data_en[5] *_debug_info_sel=18 [2:0] cfg_msix_pba_bir[2:0] [31:3] cfg_msix_pba_offset[28:0] [34:32] cfg_msix_table_bir[2:0] [63:35] cfg_msix_table_offset[28:0] [74:64] cfg_msix_table_size[10:0] *_debug_info_sel=19 [2:0] cfg_msix_pba_bir[5:3] [31:3] cfg_msix_pba_offset[57:29] [34:32] cfg_msix_table_bir[5:3] [63:35] cfg_msix_table_offset[57:29] [74:64] cfg_msix_table_size[21:11] *_debug_info_sel=20 [2:0] cfg_vf_msix_pba_bir[2:0] [31:3] cfg_vf_msix_pba_offset[28:0] [34:32] cfg_vf_msix_table_bir[2:0] [63:35] cfg_vf_msix_table_offset[28:0] [74:64] cfg_vf_msix_table_size[10:0] *_debug_info_sel=21 [2:0] cfg_vf_msix_pba_bir[5:3] [31:3] cfg_vf_msix_pba_offset[57:29] [34:32] cfg_vf_msix_table_bir[5:3] [63:35] cfg_vf_msix_table_offset[57:29] [74:64] cfg_vf_msix_table_size[21:11] *_debug_info_sel=22 [[63:0] cxpl_debug_info[63:0] [79:64] cxpl_debug_info_ei[15:0]

Port Name	Direction	Bit width	Functional description
			[95:80] cfg_int_pin [15:0] *_debug_info_sel=23 [119:0] cdm_ras_des_sd_info_v0[239:120] *_debug_info_sel=24 [119:0] cdm_ras_des_sd_info_v0[119:0] *_debug_info_sel=25 [14:0] diag_status_bus[14:0] [17:15] diag_status_bus[514:512] [37:18] diag_status_bus[817:798] *_debug_info_sel=26 [142:0] diag_status_bus[157:15] *_debug_info_sel=27 [140:0] diag_status_bus[298:158] *_debug_info_sel=28 [40:0] diag_status_bus[339:299] [44:41] diag_status_bus[471:468] [48:45] diag_status_bus[484:481] [71:49] diag_status_bus[511:489] *_debug_info_sel=29 [127:0] diag_status_bus[467:340] [136:128] diag_status_bus[480:472] [140:137] diag_status_bus[488:485] *_debug_info_sel=30 [1:0] diag_status_bus[531:530] [130:2] diag_status_bus[797:669] *_debug_info_sel=31 [133:0] diag_status_bus[668:535] *_debug_info_sel=32 [12:0] diag_status_bus[830:818] [84:13] diag_status_bus[930:859] *_debug_info_sel=33 [71:0] cfg_rbar_size[71:0]
CFG_FLR_VF_ACTIVE	O	6	Set when the software initiates FLR a virtual function by writing to the 'Initiate FLR' register bit of that function. This signal is held asserted until the reset of both the application and internal controller logic has been completed. Each bit position corresponds to one of the NVF virtual functions.

Port Name	Direction	Bit width	Functional description
CFG_FLR_PF_ACTIVE	O	2	Set when the software initiates FLR a physical function by writing to the "Initiate FLR" register bit of that function
APP_FLR_VF_DONE	I	6	Indicates that FLR a virtual function has been completed
APP_FLR_PF_DONE	I	2	Indicates that FLR a physical function has been completed
XADM_PH_CDTS	O	8	The amount of posted header buffer space currently available at the receiver at the other end of the link (in units of posted header credits
XADM_PD_CDTS	O	12	The amount of posted data buffer space currently available at the receiver at the other end of the link (in units of posted data credits).
XADM_NPH_CDTS	O	8	The amount of non-posted header buffer space currently available at the receiver at the other end of the link (in units of non-posted header credits
XADM_NPD_CDTS	O	12	The amount of non-posted data buffer space currently available at the receiver at the other end of the link (in units of non-posted data credits
XADM_CPLH_CDTS	O	8	The amount of completion header buffer space currently available at the receiver at the other end of the link (in units of completion header credits).
XADM_CPLD_CDTS	O	12	The amount of completion data buffer space currently available at the receiver at the other end of the link (in units of completion data credits)
MAC_PHY_POWERDOWN	O	2	Power control bits to the PHY 00: P0 (L0): normal 01: P0s (L0s): low recovery time, power saving. 10: P1 (L1): longer recovery time, additional power saving. 11: P2 (L2): lowest power state.
PHY_MAC_RXELECIDLE	I	8	Indicates receiver detection of an Electrical Idle for each lane
PHY_MAC_PHYSTATUS	I	8	Communicates completion of PHY functions, including power management transitions, receiver detection, speed change.
PHY_MAC_RXDATA	I	256	Parallel received data, 32 bits per lane
PHY_MAC_RXDATAK	I	32	Control (K character) indicator bits for received data
PHY_MAC_RXVALID	I	8	Indicates symbol lock and valid data for each lane
PHY_MAC_RXSTATUS	I	24	Receive status and error codes for each lane(3bits per lane)
MAC_PHY_TXDATA	O	256	Parallel data for transmission
MAC_PHY_TXDATAK	O	32	Control (K character) indicator bits for transmitted data:
MAC_PHY_TXDATAVALID	O	8	This signal allows the PCIe controller to instruct the PHY to ignore the data interface for one or two clock cycles. A value of one indicates that the data is valid and should be used
MAC_PHY_TXSYNCHEADER	O	16	At 8.0 GT/s this is used by the PCIe controller MAC to tell the PHY the value of the Sync header in the current 130b block.
MAC_PHY_TXSTARTBLOCK	O	8	Only used at the 8.0 GT/s signaling rate. This signal allows the PCIe controller MAC to indicate (to the PHY) the starting cycle mac_phy_txdata for a 128b block.
MAC_PHY_ BLOCKALIGNCONTROL	O	1	Enable EIEOS detection in the PHY
MAC_PHY_ TXDETECTRX_LOOPBACK	O	8	Combined loopback and transmit detect control

Port Name	Direction	Bit width	Functional description
MAC_PHY_TXELECIDLE_H	O	8	Forces transmit output to Electrical Idle for each lane which it is asserted,higher 16bit
MAC_PHY_TXELECIDLE_L	O	8	Forces transmit output to Electrical Idle for each lane which it is asserted,lower 16bit
MAC_PHY_TXCOMPLIANCE	O	8	Sets the running disparity to negative
MAC_PHY_RXPOLARITY	O	8	Directs the PHY to perform a polarity inversion the received data the specified lanes
MAC_PHY_RATE	O	2	Controls the link signaling rate 0: 2.5 GT/s signaling rate 1: 5.0 GT/s signaling rate 2: 8.0 GT/s signaling rate for PCIe mode or ESM data rate0 for ESM mode
MAC_PHY_TXDEEMPH	O	49	Bit 48 : mac_phy_txdeemp_valid, align with C-1 in gen3 mode. Bit 47:0 in gen3 mode : Use 3 consecutive cycles to represent C-1, C0, C+1 successively. where C-1 and mac_phy_txdeemph_valid alignment in_gen2/gen1_mode: When mac_phy_txdeemph_valid is valid, 00: -6DB de-emphasis 01: -3.5DB de-emphasis
MAC_PHY_TXMARGIN	O	3	Selects transmitter voltage levels: 000: Normal operating range 001: 800-1200 mV for Full swing* or 400-700mV for Half swing 010: Required and vendor defined 011: Required and vendor defined 100: Required and 200-400 mV for Full swing* or 100-200 mV for Half swing* if the last value or vendor defined 101: Optional and 200-400 mV for Full swing* or 100-200 mV for Half swing* if the last value or vendor defined or Reserved if no other values supported 110: Optional and 200-400 mV for Full swing* or 100-200 mV for Half swing* if the last value or vendor defined or Reserved if no other values supported 111: optional and 200-400 mV for Full swing* or 100-200 mV for Half swing* if the last value or Reserved if no other values supported *PIPE2 implementations that only support 2.5GT/s signaling rate do not implement this signal.
MAC_PHY_TXSWING	O	1	Controls the PHY transmitter voltage swing level: 0: Full swing 1: Low swing
PHY_MAC_RXDATAVALID	I	8	This signal allows a PHY to indicate valid data on the PIPE interface. A value of '1' indicates that the data is valid and should be used
PHY_MAC_RXSTARTBLOCK	I	8	Only used at the 8.0 GT/s signaling rate. This signal allows the PHY to indicate (to the PCIe controller MAC) the starting cycle phy_mac_rxdata for a 128b block. The starting byte for a 128b block must always start with Bit 0 of the data interface

Port Name	Direction	Bit width	Functional description
PHY_MAC_RXSYNCHEADER	I	16	At 8.0 GT/s this is used by the PHY to tell the PCIe controller MAC the value of the Sync header in the current 130b block
PHY_MAC_LOCALFS	I	48	At 8.0 GT/s rate, it is the Local FS value advertised by the local PHY.
PHY_MAC_LOCALLF	I	48	At 8.0 GT/s rate, it is the Local LF value advertised by the local PHY.
MAC_PHY_RXPRESETHINT	O	24	Provides the RX equalization preset hint for the receiver.
MAC_PHY_FS	O	48	Provides the FS value advertised by the link partner.
MAC_PHY_LF	O	48	Provides the LF value advertised by the link partner.
PHY_MAC_DIRFEEDBACK	I	48	Provides the link equalization evaluation feedback in the Direction Change format. Feedback is provided for each coefficient: [1:0]: C-1 [3:2]: C0 [5:4]: C+1 The feedback value for each coefficient is encoded as follows: 00: No change 01: Increment by 1 10: Decrement by 1 11: Reserved
PHY_MAC_FOMFEEDBACK	I	64	Provides the link equalization evaluation feedback in the Figure of Merit (FOM) format
PHY_MAC_LOCAL_TX_ PSET_COEF	I	48	Use three consecutive cycles to represent C-1, C0, C+1, and other phy_mac_local_tx_coef_valid corresponds to C+1
PHY_MAC_LOCAL_TX_ COEF_VALID	I	8	Single pulse indicating that the phy_mac_local_tx_pset_coef bus correctly represents the coefficients values mapped from the preset the mac_phy_local_pset_index bus
MAC_PHY_LOCAL_PSET_ INDEX	O	32	The TX equalization preset that the MAC wants the PHY to convert into coefficients
MAC_PHY_GETLOCAL_ PSET_COEF	O	8	Single pulse used by MAC to request the PHY to perform a preset-to-co- efficient mapping, from preset mac_phy_local_pset_index, to coefficients phy_mac_local_tx_pset_coef
MAC_PHY_RXEQINPROGRESS	O	8	This status signal is asserted and remains asserted for the duration of Recovery.Equalization Phase2 when the PCIe controller is the USP (Phase3 when DSP).
MAC_PHY_INVALID_REQ	O	8	Indicates that the Link Evaluation feedback requested a link partner TX EQ setting that has been out of range.
MAC_PHY_RXEQEVAL	O	8	The PHY starts evaluation of the far end transmitter TX EQ settings while this signal is held high by the controller
MAC_PHY_DIRCHANGE	O	8	Indicates the PHY to perform Figure Of Merit or Direction Change evaluation
XDLH_RETRYRAM_ADDR	O	10	Retry buff read and write address
XDLH_RETRYRAM_DATA	O	135	Retry buff write data
XDLH_RETRYRAM_WE	O	1	Retry buff read and write enable
XDLH_RETRYRAM_EN	O	1	Retry buff read and write enable

Port Name	Direction	Bit width	Functional description
RETRYRAM_XDLH_DATA	I	135	retry buff read data
PNP_RAM_RD_DATA	I	135	rcv ram read data
PNP_RAM_WR_ADDR	O	10	rcv ram write address
PNP_RAM_RD_ADDR	O	10	rcv ram read address
PNP_RAM_WR_DATA	O	135	rcv ram write data
PNP_RAM_WR_EN	O	1	rcv ram write enable
PNP_RAM_RD_EN	O	1	rcv ram read enable
RAM_TEST_EN	I	1	external_ram test enable, active high
RAM_TEST_ADDRH	I	1	Test mode: 1: test addr is in the high position of the data 0: test addr is in the low bit of data
RETRY_TEST_DATA_EN	I	1	Test mode (only for single port ram): 1: Test data line 0: test control line
RAM_TEST_MODE_N	I	1	external_ram test mode, active low 1: normal mode 0: ram test mode
TPH_RAM_ADDR	O	5	Single port Ram read and write address
TPH_RAM_FUNC_NUM	O	3	0:pf0 1: pf1 2:vf 0 …. 7:vf6
TPH_RAM_FUNC_ACTIVE	O	1	High effective, indicating that the current func_num is valid
TPH_RAM_WR_DATA	O	16	Single port ram write data
TPH_RAM_WR_BYTE_EN	O	2	Single port ram write byte_en
TPH_RAM_WR_EN	O	1	Single port ram read and write enable. 1: write valid 0: read valid
TPH_RAM_RD_DATA	I	16	Single port ram read data

Note: The above ports are all port names and their function descriptions in the function working mode. If it is in the DFT test mode, the following three ports will be added.

Port Name	Direction	Bit width	Functional description
TEST_SE_N	I	1	Test shift enable, active low
TEST_RST_N	I	1	Test reset, active low
TEST_MODE_N	I	1	Active low to enable test mode

4. Parameter description

Table 346 GTP_PCIEGEN3 parameter list

Parameter Name	Parameter Type	Validvalue	Defaults	Functional description
PF1_ENABLE	<string>	"FALSE", "TRUE"	TRUE	It is used to configure whether to support PF1, if it is 1, it means to support PF1.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
				This sc only modifies the next function number of the ARI register. When PF1 is not supported, it is also necessary to write bit 7 of the header type register to 0 through the DBI interface.
FLT_SHORT_TLP	<string>	"FALSE", "TRUE"	TRUE	When it is 1, the controller internal filter tlp below 3dw
APP_DEV_NUM	<integer>	0 to 31	0	Device number. Your application must drive this signal to set the device number in the Requester ID for RC mode.
APP_BUS_NUM	<integer>	0 to 255	0	Bus number. Your application must drive this signal to set the bus number in the Requester ID for RC mode.
DEBUG_INFO_SEL	<integer>	0 to 33	0	Debug output select
DYN_DEBUG_SEL_EN	<string>	"FALSE", "TRUE"	FALSE	When it is 0, debug_info_sel controls the debug_info_mux signal group selection dyn_debug_info_sel controls debug_info_mux signal group selection when 1
GRS_EN	<string>	"FALSE", "TRUE"	FALSE	Active high mask grsn input
VF_ER_REPORT_EN	<string>	"FALSE", "TRUE"	TRUE	High effective, modify the bug of VF error reporting
ATOMIC_DISABLE	<string>	"FALSE", "TRUE"	FALSE	High effective, Disable Atomic TLP function
VC_DISABLE	<string>	"FALSE", "TRUE"	FALSE	Active high, used to disable VC capability.
SRIOV_DISABLE	<string>	"FALSE", "TRUE"	FALSE	Active high, used for disableriov function
BAR_RESIZABLE	<integer>	It is related to NUM OF RBARS, see the function description column for details	1365	High effective, configure whether BAR is resizeable: bit11 : PF1 BAR5 bit10: PF1 BAR4 …… bit6:PF1 BAR0 bit5:PF0 BAR 5 …… bit0: PF0 BAR0. The default value is b0101_0101_0101

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
				BAR_RESIZABLE is related to NUM OF RBARS. For example, when NUM OF RBAR[2:0] is 011, BAR_RESIZABLE[5:0] requires any 3 bits to be 1. When NUM OF RBAR[5:3] is 010, BAR_RESIZABLE[11: 6] Need any 2bit to be 1. Others are similar
NUM_OF_RBARS	<integer>	"0, 1, 2, 3, 8, 9, 10, 11, 16, 17, 18, 19, 24, 25, 26, 27"	27	High effective, corresponding to the Number of Resizable BARs in the Resizable BAR Control Register; bit5~bit 3 for PF1, bit2~bit0 for PF0 IP The default values generated by PF1 and PF0 are 3 1 resizeable bars, PF supports up to 3 RBARS, the legal values of NUM OFRBAR[2:0] are 000, 001,010,011. The legal values of NUM OF RBAR[5:3] are 000, 001,010,011. That is, NUM OF RBAR has a total of 16 legal values
MSI_PVM_DISABLE	<integer>	"0 to 255"	0	High effective, used to disable the pending and mask functions of msi Bit7 ~bit2 :VF5~VF0Bit1~bit0:PF1~PF0
BAR_MASK_WRITABLE	<integer>	"0 to 4095"	0	High effective means that the corresponding bar_mask value can be written. bit11 : PF1 BAR5 bit10: PF1 BAR4 …… bit6:PF1 BAR0 bit5:PF0 BAR 5 …… bit0:PF0 BAR0

Note: The above parameters can be used to configure PCIE internal function modules, which can be flexibly selected

5. Instantiation template

```
GTP_PCIEGEN3 #(
.PF1_ENABLE("TRUE"),
.FLT_SHORT_TLP("TRUE"),
```

```
.APP_DEV_NUM(0),
.APP_BUS_NUM(0),
.DEBUG_INFO_SEL(0),
.DYN_DEBUG_SEL_EN("FALSE"),
.GRS_EN("FALSE"),
.VF_ER_REPORT_EN("TRUE"),
.ATOMIC_DISABLE("FALSE"),
.VC_DISABLE("FALSE"),
.SRIOV_DISABLE("FALSE"),
.BAR_RESIZABLE(1365),
.NUM_OF_RBARS(27),
.MSI_PVM_DISABLE(0),
.BAR_MASK_WRITABLE(0)
) GTP_PCIEGEN3_inst (
.AUX_PM_EN(),                    // OUTPUT[1:0]
.AXIS_MASTER0_TDATA(),           // OUTPUT[255:0]
.AXIS_MASTER0_TKEEP(),           // OUTPUT[7:0]
.AXIS_MASTER0_TUSER(),           // OUTPUT[12:0]
.AXIS_MASTER1_TDATA(),           // OUTPUT[255:0]
.AXIS_MASTER1_TKEEP(),           // OUTPUT[7:0]
.AXIS_MASTER1_TUSER(),           // OUTPUT[33:0]
.CFG_AER_RC_ERR_INT(),           // OUTPUT[1:0]
.CFG_AER_RC_ERR_MSI(),           // OUTPUT[1:0]
.CFG_ARI_FWD_EN(),               // OUTPUT[1:0]
.CFG_ATOMIC_EGRESS_BLOCK(),      // OUTPUT[1:0]
.CFG_ATOMIC_REQ_EN(),            // OUTPUT[1:0]
.CFG_BUS_MASTER_EN(),            // OUTPUT[1:0]
.CFG_COR_ERR_RPT_EN(),           // OUTPUT[1:0]
.CFG_CRS_SW_VIS_EN(),            // OUTPUT[1:0]
.CFG_EXT_TAG_EN(),               // OUTPUT[1:0]
.CFG_FLR_PF_ACTIVE(),            // OUTPUT[1:0]
.CFG_FLR_VF_ACTIVE(),            // OUTPUT[5:0]
.CFG_F_ERR_RPT_EN(),             // OUTPUT[1:0]
.CFG_IDO_CPL_EN(),               // OUTPUT[1:0]
.CFG_IDO_REQ_EN(),               // OUTPUT[1:0]
.CFG_INT_DISABLE(),              // OUTPUT[1:0]
.CFG_MAX_PAYLOAD_SIZE(),         // OUTPUT[5:0]
.CFG_MAX_RD_REQ_SIZE(),          // OUTPUT[5:0]
.CFG_MEM_SPACE_EN(),             // OUTPUT[1:0]
.CFG_MSIX_EN(),                  // OUTPUT[1:0]
.CFG_MSIX_FUNC_MASK(),           // OUTPUT[1:0]
.CFG_MSI_EN(),                   // OUTPUT[1:0]
.CFG_MULTI_MSI_EN(),             // OUTPUT[5:0]
.CFG_NF_ERR_RPT_EN(),            // OUTPUT[1:0]
.CFG_NO_SNOOP_EN(),              // OUTPUT[1:0]
.CFG_NUM_VF(),                   // OUTPUT[5:0]
.CFG_OBFF_EN(),                  // OUTPUT[1:0]
.CFG_PBUS_DEV_NUM(),             // OUTPUT[4:0]
```

```
    .CFG_PBUS_NUM(),                  // OUTPUT[7:0]
    .CFG_PF_TPH_ST_MODE(),            // OUTPUT[5:0]
    .CFG_PME_INT(),                   // OUTPUT[1:0]
    .CFG_PME_MSI(),                   // OUTPUT[1:0]
    .CFG_PWR_BUDGET_DATA_SEL_REG(), // OUTPUT[7:0]
    .CFG_PWR_BUDGET_SEL(),            // OUTPUT[1:0]
    .CFG_RCB(),                       // OUTPUT[1:0]
    .CFG_REG_SERREN(),                // OUTPUT[1:0]
    .CFG_RELAX_ORDER_EN(),            // OUTPUT[1:0]
    .CFG_SEND_COR_ERR(),              // OUTPUT[1:0]
    .CFG_SEND_F_ERR(),                // OUTPUT[1:0]
    .CFG_SEND_NF_ERR(),               // OUTPUT[1:0]
    .CFG_START_VFI(),                 // OUTPUT[5:0]
    .CFG_SYS_ERR_RC(),                // OUTPUT[1:0]
    .CFG_TC_ENABLE(),                 // OUTPUT[7:0]
    .CFG_TPH_REQ_EN(),                // OUTPUT[1:0]
    .CFG_VF_BME(),                    // OUTPUT[5:0]
    .CFG_VF_EN(),                     // OUTPUT[1:0]
    .CFG_VF_MSIX_EN(),                // OUTPUT[5:0]
    .CFG_VF_MSIX_FUNC_MASK(),         // OUTPUT[5:0]
    .CFG_VF_MSI_EN(),                 // OUTPUT[5:0]
    .CFG_VF_MULTI_MSI_EN(),           // OUTPUT[17:0]
    .CFG_VF_TPH_REQ_EN(),             // OUTPUT[5:0]
    .CFG_VF_TPH_ST_MODE(),            // OUTPUT[17:0]
    .CORE_AVL_NP_CNT(),               // OUTPUT[5:0]
    .CORE_AVL_P_CNT(),                // OUTPUT[5:0]
    .DEBUG_INFO_MUX(),                // OUTPUT[142:0]
    .DPA_SUBSTATE_UPDATE(),           // OUTPUT[1:0]
    .LBC_DBI_DOUT(),                  // OUTPUT[31:0]
    .MAC_PHY_DIRCHANGE(),             // OUTPUT[7:0]
    .MAC_PHY_FS(),                    // OUTPUT[47:0]
    .MAC_PHY_GETLOCAL_PSET_COEF(),   // OUTPUT[7:0]
    .MAC_PHY_INVALID_REQ(),           // OUTPUT[7:0]
    .MAC_PHY_LF(),                    // OUTPUT[47:0]
    .MAC_PHY_LOCAL_PSET_INDEX(),      // OUTPUT[31:0]
    .MAC_PHY_POWERDOWN(),             // OUTPUT[1:0]
    .MAC_PHY_RATE(),                  // OUTPUT[1:0]
    .MAC_PHY_RXEQEVAL(),              // OUTPUT[7:0]
    .MAC_PHY_RXEQINPROGRESS(),        // OUTPUT[7:0]
    .MAC_PHY_RXPOLARITY(),            // OUTPUT[7:0]
    .MAC_PHY_RXPRESETHINT(),          // OUTPUT[23:0]
    .MAC_PHY_TXCOMPLIANCE(),          // OUTPUT[7:0]
    .MAC_PHY_TXDATA(),                // OUTPUT[255:0]
    .MAC_PHY_TXDATAK(),               // OUTPUT[31:0]
    .MAC_PHY_TXDATAVALID(),           // OUTPUT[7:0]
    .MAC_PHY_TXDEEMPH(),              // OUTPUT[48:0]
    .MAC_PHY_TXDETECTRX_LOOPBACK(), // OUTPUT[7:0]
    .MAC_PHY_TXELECIDLE_H(),          // OUTPUT[7:0]
```

```
.MAC_PHY_TXELECIDLE_L(),          // OUTPUT[7:0]
.MAC_PHY_TXMARGIN(),              // OUTPUT[2:0]
.MAC_PHY_TXSTARTBLOCK(),          // OUTPUT[7:0]
.MAC_PHY_TXSYNCHEADER(),          // OUTPUT[15:0]
.MSG_RCVD_DATA(),                 // OUTPUT[7:0]
.MSG_RCVD_TYPE(),                 // OUTPUT[4:0]
.PM_DSTATE(),                     // OUTPUT[5:0]
.PM_MASTER_STATE(),               // OUTPUT[4:0]
.PM_PME_EN(),                     // OUTPUT[1:0]
.PM_SLAVE_STATE(),                // OUTPUT[4:0]
.PM_STATUS(),                     // OUTPUT[1:0]
.PM_VF_DSTATE(),                  // OUTPUT[17:0]
.PM_VF_PME_EN(),                  // OUTPUT[5:0]
.PM_VF_STATUS(),                  // OUTPUT[5:0]
.PNP_RAM_RD_ADDR(),               // OUTPUT[9:0]
.PNP_RAM_WR_ADDR(),               // OUTPUT[9:0]
.PNP_RAM_WR_DATA(),               // OUTPUT[134:0]
.RADM_CPL_TIMEOUT(),              // OUTPUT[1:0]
.RADM_TIMEOUT_CPL_ATTR(),         // OUTPUT[3:0]
.RADM_TIMEOUT_CPL_LEN(),          // OUTPUT[21:0]
.RADM_TIMEOUT_CPL_TAG(),          // OUTPUT[15:0]
.RADM_TIMEOUT_CPL_TC(),           // OUTPUT[5:0]
.RADM_TIMEOUT_FUNC_NUM(),         // OUTPUT[1:0]
.RADM_TIMEOUT_VFUNC_ACTIVE(),     // OUTPUT[1:0]
.RADM_TIMEOUT_VFUNC_NUM(),        // OUTPUT[5:0]
.RBAR_CTRL_UPDATE(),              // OUTPUT[1:0]
.SEDO(),                          // OUTPUT[1:0]
.SEDO_EN(),                       // OUTPUT[1:0]
.SMLH_LTSSM_STATE(),              // OUTPUT[5:0]
.TPH_RAM_ADDR(),                  // OUTPUT[4:0]
.TPH_RAM_FUNC_NUM(),              // OUTPUT[2:0]
.TPH_RAM_WR_BYTE_EN(),            // OUTPUT[1:0]
.TPH_RAM_WR_DATA(),               // OUTPUT[15:0]
.XADM_CPLD_CDTS(),                // OUTPUT[11:0]
.XADM_CPLH_CDTS(),                // OUTPUT[7:0]
.XADM_NPD_CDTS(),                 // OUTPUT[11:0]
.XADM_NPH_CDTS(),                 // OUTPUT[7:0]
.XADM_PD_CDTS(),                  // OUTPUT[11:0]
.XADM_PH_CDTS(),                  // OUTPUT[7:0]
.XDLH_RETRYRAM_ADDR(),            // OUTPUT[9:0]
.XDLH_RETRYRAM_DATA(),            // OUTPUT[134:0]
.APPS_PM_VF_XMT_PME(),            // INPUT[5:0]
.APPS_PM_XMT_PME(),               // INPUT[1:0]
.APP_ERR_BUS(),                   // INPUT[12:0]
.APP_ERR_VFUNC_NUM(),             // INPUT[2:0]
.APP_FLR_PF_DONE(),               // INPUT[1:0]
.APP_FLR_VF_DONE(),               // INPUT[5:0]
.APP_HDR_LOG(),                   // INPUT[127:0]
```

```
.APP_LTR_MSG_LATENCY(),          // INPUT[31:0]
.APP_PF_REQ_RETRY_EN(),          // INPUT[1:0]
.APP_VF_REQ_RETRY_EN(),          // INPUT[5:0]
.AXIS_SLAVE0_TDATA(),            // INPUT[255:0]
.AXIS_SLAVE0_TUSER(),            // INPUT[5:0]
.AXIS_SLAVE1_TDATA(),            // INPUT[255:0]
.AXIS_SLAVE1_TUSER(),            // INPUT[5:0]
.AXIS_SLAVE2_TDATA(),            // INPUT[255:0]
.AXIS_SLAVE2_TUSER(),            // INPUT[5:0]
.CFG_MSI_PENDING(),              // INPUT[63:0]
.CFG_PWR_BUDGET_DATA_REG(),      // INPUT[31:0]
.CFG_VF_MSI_PENDING(),           // INPUT[191:0]
.DBI_ADDR(),                     // INPUT[31:0]
.DBI_DIN(),                      // INPUT[31:0]
.DBI_VFUNC_NUM(),                // INPUT[2:0]
.DBI_WR(),                       // INPUT[3:0]
.DEVICE_TYPE(),                  // INPUT[3:0]
.DIAG_CTRL_BUS(),                // INPUT[2:0]
.DYN_DEBUG_INFO_SEL(),           // INPUT[5:0]
.MSIX_ADDR(),                    // INPUT[63:0]
.MSIX_DATA(),                    // INPUT[31:0]
.PHY_MAC_DIRFEEDBACK(),          // INPUT[47:0]
.PHY_MAC_FOMFEEDBACK(),          // INPUT[63:0]
.PHY_MAC_LOCALFS(),              // INPUT[47:0]
.PHY_MAC_LOCALLF(),              // INPUT[47:0]
.PHY_MAC_LOCAL_TX_COEF_VALID(), // INPUT[7:0]
.PHY_MAC_LOCAL_TX_PSET_COEF(),   // INPUT[47:0]
.PHY_MAC_PHYSTATUS(),            // INPUT[7:0]
.PHY_MAC_RXDATA(),               // INPUT[255:0]
.PHY_MAC_RXDATAK(),              // INPUT[31:0]
.PHY_MAC_RXDATAVALID(),          // INPUT[7:0]
.PHY_MAC_RXELECIDLE(),           // INPUT[7:0]
.PHY_MAC_RXSTARTBLOCK(),         // INPUT[7:0]
.PHY_MAC_RXSTATUS(),             // INPUT[23:0]
.PHY_MAC_RXSYNCHEADER(),         // INPUT[15:0]
.PHY_MAC_RXVALID(),              // INPUT[7:0]
.PNP_RAM_RD_DATA(),              // INPUT[134:0]
.RETRYRAM_XDLH_DATA(),           // INPUT[134:0]
.SEDI(),                         // INPUT[1:0]
.SEDI_ACK(),                     // INPUT[1:0]
.SYS_INT(),                      // INPUT[1:0]
.TPH_RAM_RD_DATA(),              // INPUT[15:0]
.VEN_MSI_TC(),                   // INPUT[2:0]
.VEN_MSI_VECTOR(),               // INPUT[4:0]
.VEN_MSI_VFUNC_NUM(),            // INPUT[2:0]
.APP_LTR_MSG_GRANT(),            // OUTPUT
.APP_OBFF_MSG_GRANT(),           // OUTPUT
.AXIS_MASTER0_TLAST(),           // OUTPUT
```

```
.AXIS_MASTER0_TVALID(),          // OUTPUT
.AXIS_MASTER1_TVALID(),          // OUTPUT
.AXIS_SLAVE0_TREADY(),           // OUTPUT
.AXIS_SLAVE1_TREADY(),           // OUTPUT
.AXIS_SLAVE2_TREADY(),           // OUTPUT
.CFG_2ND_RESET(),                // OUTPUT
.CFG_BW_MGT_INT(),               // OUTPUT
.CFG_BW_MGT_MSI(),               // OUTPUT
.CFG_DISABLE_LTR_CLR_MSG(),      // OUTPUT
.CFG_LINK_AUTO_BW_INT(),         // OUTPUT
.CFG_LINK_AUTO_BW_MSI(),         // OUTPUT
.CFG_LINK_EQ_REQ_INT(),          // OUTPUT
.CFG_LTR_M_EN(),                 // OUTPUT
.CFG_MSI_MASK_UPDATE(),          // OUTPUT
.INT_GRT(),                      // OUTPUT
.LBC_DBI_ACK(),                  // OUTPUT
.LINK_REQ_RST(),                 // OUTPUT
.MAC_PHY_BLOCKALIGNCONTROL(),    // OUTPUT
.MAC_PHY_TXSWING(),              // OUTPUT
.MSG_RCVD(),                     // OUTPUT
.PHY_RST_N(),                    // OUTPUT
.PM_XTLH_BLOCK_TLP(),            // OUTPUT
.PNP_RAM_RD_EN(),                // OUTPUT
.PNP_RAM_WR_EN(),                // OUTPUT
.RADM_QOVERFLOW(),               // OUTPUT
.RADM_Q_NOT_EMPTY(),             // OUTPUT
.RDLH_LINK_UP(),                 // OUTPUT
.SMLH_LINK_UP(),                 // OUTPUT
.SMLH_REQ_RST(),                 // OUTPUT
.TPH_RAM_FUNC_ACTIVE(),          // OUTPUT
.TPH_RAM_WR_EN(),                // OUTPUT
.TRAINING_RST_N(),               // OUTPUT
.USER_RST_N(),                   // OUTPUT
.VEN_MSI_GRANT(),                // OUTPUT
.WAKE(),                         // OUTPUT
.XDLH_RETRYRAM_EN(),             // OUTPUT
.XDLH_RETRYRAM_WE(),             // OUTPUT
.APPS_PM_XMT_TURNOFF(),          // INPUT
.APP_CLK_PM_EN(),                // INPUT
.APP_DBI_RO_WR_DISABLE(),        // INPUT
.APP_ERR_ADVISORY(),             // INPUT
.APP_ERR_FUNC_NUM(),             // INPUT
.APP_ERR_VFUNC_ACTIVE(),         // INPUT
.APP_HDR_VALID(),                // INPUT
.APP_INIT_RST(),                 // INPUT
.APP_LTR_MSG_FUNC_NUM(),         // INPUT
.APP_LTR_MSG_REQ(),              // INPUT
.APP_LTSSM_ENABLE(),             // INPUT
```

```
.APP_OBFF_CPU_ACTIVE_MSG_REQ(), // INPUT
.APP_OBFF_IDLE_MSG_REQ(),       // INPUT
.APP_OBFF_OBFF_MSG_REQ(),       // INPUT
.APP_RAS_DES_SD_HOLD_LTSSM(),   // INPUT
.APP_READY_ENTR_L23(),          // INPUT
.APP_REQ_ENTR_L1(),             // INPUT
.APP_REQ_EXIT_L1(),             // INPUT
.APP_REQ_RETRY_EN(),            // INPUT
.APP_UNLOCK_MSG(),              // INPUT
.APP_XFER_PENDING(),            // INPUT
.AXIS_MASTER0_TREADY(),         // INPUT
.AXIS_SLAVE0_TLAST(),           // INPUT
.AXIS_SLAVE0_TVALID(),          // INPUT
.AXIS_SLAVE1_TLAST(),           // INPUT
.AXIS_SLAVE1_TVALID(),          // INPUT
.AXIS_SLAVE2_TLAST(),           // INPUT
.AXIS_SLAVE2_TVALID(),          // INPUT
.BUTTON_RST(),                  // INPUT
.CFG_PWR_BUDGET_FUNC_NUM(),     // INPUT
.CFG_PWR_BUDGET_VALID(),        // INPUT
.DBI_CS(),                      // INPUT
.DBI_CS2(),                     // INPUT
.DBI_FUNC_NUM(),                // INPUT
.DBI_VFUNC_ACTIVE(),            // INPUT
.MEM_CLK(),                     // INPUT
.PCLK(),                        // INPUT
.PCLK_DIV2(),                   // INPUT
.PERST(),                       // INPUT
.POWER_UP_RST(),                // INPUT
.RAM_TEST_ADDRH(),              // INPUT
.RAM_TEST_EN(),                 // INPUT
.RAM_TEST_MODE_N(),             // INPUT
.RETRY_TEST_DATA_EN(),          // INPUT
.RX_LANE_FLIP_EN(),             // INPUT
.TPH_RD_DATA_VALID(),           // INPUT
.TX_LANE_FLIP_EN(),             // INPUT
.USER_CLK(),                    // INPUT
.USER_RCVD_NP_READY(),          // INPUT
.USER_RCVD_P_READY(),           // INPUT
.VEN_MSI_FUNC_NUM(),            // INPUT
.VEN_MSI_REQ(),                 // INPUT
.VEN_MSI_VFUNC_ACTIVE()         // INPUT
);
```

(7) GTP_HSSTHP_LANE instruction

1. Supported devices

Table 347 GTP_HSSTHP_LANE device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

HSSTHP_LANE_TOP includes the following submodules:

1. One lane of PCS:pcs13g_top_pcs_single.

2. Two clk dly line modules: dqsl_ddr_dll_dly_var_2x.

3. Two phase detect modules: pma_phase_detect.

4. A lane of PMA: pma_channel (with LPLL inside) HSSTHP_LANE exists as a prim, and add parameter to sc conversion on the basis of HSSTHP_LANE_TOP to form GTP_HSSTHP_LANE, so that HSSTHP_LANE can be easily converted to GTP_HSSTHP_LANE (satisfies "GTP coding specification.doc"). HSSTHP_LANE contains an LPLL (working range is 1.6GHz~6.6GHz), users can use it in low-speed scenarios. For detailed functions, please refer to "UG050008_Titan2 Series FPGA High-Speed Serial Transceiver (HSSTHP) User Guide".

3. Port description

Table 348 GTP_HSSTHP_LANE port list

Port Name	Direction	Data width	Functional description
P_CFG_READY	O	1	Read and write ready output of dynamic configuration interface, high effective
P_CFG_RDATA	O	8	Read data of dynamic configuration interface
P_CFG_INT	O	1	Interrupt output of dynamic configuration interface, active high
LANE_COUT_BUS_FORWARD	O	25	Cascade signal, the transmission direction is the COUT port bus of LANE0->LANE1->LANE2->LANE3, See Table 349 for the description of each segment
LANE_COUT_BUS_BACKWARD	O	2	Cascade signal, the transmission direction is the COUT port bus of LANE3->LANE2->LANE1->LANE0: [0] corresponds to CB_FIFO_STATE_COUT; [1] corresponds to APATTERN_STATUS_COUT
P_RX_PRBS_ERROR	O	1	Rx prbs check error, async signal
P_PCS_RX_MCB_STATUS	O	1	Channel bonding status indicator signal, if it is 1, it means the bonding is successful
P_PCS_LSM_SYNCED	O	1	Word Alignment Alignment flag signal, state machine locked, active high, asynchronous signal
P_RDATA	O	88	Parallel data received by the user
P_RXDVLD	O	1	receive data[31:0] valid of channel

Port Name	Direction	Data width	Functional description
P_RXDVLD_H	O	1	receive data[63:32] valid of channel
P_RXSTATUS	O	6	Encodes receiver status and error codes for the received data stream when receiving data
P_EM_ERROR_CNT	O	3	error count(gray code)
P_LPLL_READY	O	1	lane pll lock flag signal
P_RX_SIGDET_STATUS	O	1	rx signal detect status to core
P_RX_SATA_COMINIT	O	1	SATA protocol related signals
P_RX_SATA_COMWAKE	O	1	SATA protocol related signals
P_RX_LS_DATA	O	1	rx output low speed data
P_RX_READY	O	1	The ready signal of rx
P_TEST_STATUS	O	20	rx output test status register
P_TX_RXDET_STATUS	O	1	The output status signal of tx_rxdet, receiver detection status to core
P_RCLK2FABRIC	O	1	rclk to CORE
P_TCLK2FABRIC	O	1	tclk to CORE
P_CA_ALIGN_RX	O	1	Receive Channel CLK Aligner dynamic status output, 0->1 transition status indicates aligner success, which is an asynchronous signal.
P_CA_ALIGN_TX	O	1	Send ChannelAligner dynamic status output, the transition status of 0 > 1 indicates that the aligner is successful, which is an asynchronous signal.
P_TX_SDN	O	1	tx output pin, N polarity
P_TX_SDP	O	1	tx output pin, P polarity
P_RX_CLK_FR_CORE	I	1	rclk of channel, from CORE
P_RCLK2_FR_CORE	I	1	from CORE, generated by refck2core through PLL multiplication
P_TX_CLK_FR_CORE	I	1	tclk of channel,from CORE
P_TCLK2_FR_CORE	I	1	from CORE, generated by refck2core through PLL multiplication
P_PCS_RX_RST	I	1	Reset signal from fabric, reset RX PCS logic, high level reset
P_PCS_TX_RST	I	1	Reset signal from fabric, reset TX PCS logic, high level reset
P_EXT_BRIDGE_PCS_RST	I	1	External bridge unit reset, async
P_CFG_RST	I	1	The reset signal of the dynamic configuration interface, high level reset. All registers are restored to the initial values set by Parameter after reset
P_CFG_CLK	I	1	Clock signal for dynamic configuration interface
P_CFG_PSEL	I	1	The selection signal of the dynamic configuration interface, active high
P_CFG_ENABLE	I	1	Access enable of dynamic configuration interface, high effective
P_CFG_WRITE	I	1	The read and write selection signal of the dynamic configuration interface, high level means write, low level means read
P_CFG_ADDR	I	12	Address bus for dynamically configured interfaces
P_CFG_WDATA	I	8	Write data of dynamic configuration interface
LANE_CIN_BUS_FORWARD	I	25	Cascade signal, the transmission direction is the CIN port bus of LANE0->LANE1->LANE2->LANE3, See Table 350 for the description of each segment
LANE_CIN_BUS_BACKWARD	I	2	Cascade signal, the transmission direction is the CIN port bus of LANE3->LANE2->LANE1->LANE0: [0] corresponds to CB_FIFO_STATE_CIN; [1] corresponds to APATTERN_STATUS_CIN

Port Name	Direction	Data width	Functional description
P_TDATA	I	88	Parallel data sent by user
P_PCIE_EI_H	I	1	Forces tx output to electrical idle,for txd[31:16]
P_PCIE_EI_L	I	1	Forces tx output to electrical idle,for txd[15:0]
P_TX_DEEMP	I	16	transmitter de-emphasis from core [15:11] : c+1 [10:5] : c0 [4:0] : c-1
P_TX_DEEMP_POST_SEL	I	2	transmitter de-emphasis post select
P_BLK_ALIGN_CTRL	I	1	Controls whether the phy performs block alignment.
P_TX_ENC_TYPE	I	1	0: enable 8b/10b encode 1: enable 128b/130b encode
P_RX_DEC_TYPE	I	1	0: enable 8b/10b decode 1: enable 128b/130b decode
P_PCS_BIT_SLIP	I	1	slip indication to word align
P_PCS_WORD_ALIGN_EN	I	1	align_en from CORE, for word align,used when word align working at outside state machine or CLK SLIP MODE
P_RX_POLARITY_INVERT	I	1	The polarity inversion enable of Rx Sample Reg is an asynchronous signal, active high, one for each Channel: P_RX_POLARITY_INVERT[0]: For Channel 0 P_RX_POLARITY_INVERT[1]: For Channel 1 P_RX_POLARITY_INVERT[2]: For Channel 2 P_RX_POLARITY_INVERT[3]: For Channel 3
P_PCS_MCB_EXT_EN	I	1	Channel bonding outside state machine enablesignals, for asynchronous signals, high effective
P_PCS_NEAREND_LOOP	I	1	pcs near end loopback ctrl signals Note: The internal clock selection register will be configured accordingly when enabled;
P_PCS_FAREND_LOOP	I	1	pcs far end loopback ctrl signals Note: Internal clock selection register configuration is required when enabling;
P_PMA_NEAREND_PLOOP	I	1	Logic or with reg_rx_tx2rx_plpbk_en to pma, PMA near-end parallel loopback control
P_PMA_NEAREND_SLOOP	I	1	Logic or with reg_tx_tx2rx_slpback_en to pma, PMA near-end serial loopback control
P_PMA_FAREND_PLOOP	I	1	Logic or with reg_tx_data_mux_sel to pma, PMA remote parallel loopback control
P_PCS_PRBS_EN	I	1	Enable tx prbs generator
P_LANE_POWERDOWN	I	1	power down lane
P_LANE_RST	I	1	reset lane, high active
P_RX_LANE_POWERDOWN	I	1	rx lane powerdown from core
P_RX_PMA_RST	I	1	Reset the PMA Receiver, which is an asynchronous signal
P_RX_CDR_RST	I	1	reset lane rx cdr, high active
P_RX_CLKPATH_RST	I	1	reset lane rx clk patn, high active
P_RX_DFE_RST	I	1	reset lane rx dfe, high active
P_RX_LEQ_RST	I	1	reset lane rx leq, high active
P_RX_SLIDING_RST	I	1	reset lane rx sliding, high active

Port Name	Direction	Data width	Functional description
P_RX_DFE_EN	I	1	lane rx dfe enable
P_RX_T1_EN	I	1	lane rx dfe t1 buffer enable
P_RX_CDRX_EN	I	1	lane rx cdrx enable
P_RX_T1_DFE_EN	I	1	lane rx t1 dfe enable
P_RX_T2_DFE_EN	I	1	lane rx t2 dfe enable
P_RX_T3_DFE_EN	I	1	lane rx t3 dfe enable
P_RX_T4_DFE_EN	I	1	lane rx t4 dfe enable
P_RX_T5_DFE_EN	I	1	lane rx t5 dfe enable
P_RX_T6_DFE_EN	I	1	lane rx t6 dfe enable
P_RX_SLIDING_EN	I	1	lane rx sliding dfe enable
P_RX_EYE_RST	I	1	reset rx eye clock path, high active
P_RX_EYE_EN	I	1	rx eye monitor buffer enable,from fabric to pma
P_RX_EYE_TAP	I	8	rx eye monitor tap,from fabric to pma
P_RX_PIC_EYE	I	8	rx eye monitor PI control,from fabric to pma
P_RX_PIC_FASTLOCK	I	8	rx fastlock PI control,from fabric to pma
P_RX_PIC_FASTLOCK_STROBE	I	1	rx fastlock PI control strobe, fabric to pma
P_EM_RD_TRIGGER	I	1	1: Enable capturing data immediately 0: disable
P_EM_MODE_CTRL	I	2	control signal for executing mode
P_RX_CTLE_DCCAL_RST	I	1	ctle dc offset reset, high active
P_RX_SLICER_DCCAL_RST	I	1	slicer dc offset reset, high active
P_RX_SLICER_DCCAL_EN	I	1	rx slicer dc offset enable
P_RX_CTLE_DCCAL_EN	I	1	rx ctle dc offset enable
P_RX_SLIP_RST	I	1	rx slip reset, high active
P_RX_SLIP_EN	I	1	rx slip enable
P_LPLL_POWERDOWN	I	1	lpll powerdown
P_LPLL_RST	I	1	lpll reset, high active
P_LPLL_LOCKDET_RST	I	1	lane pll lock detecter rst, high active
P_TX_LS_DATA	I	1	tx low speed data from core
P_TX_BEACON_EN	I	1	When active high and EI TX is enabled, send beacon enable signal
P_TX_SWING	I	1	tx_swing control signal
P_TX_RXDET_REQ	I	1	tx_rxdet_req signal
P_TX_RATE	I	2	TX rate control signal; 2'b00: data rate is 1/4 of the tx input high requency clock 2'b01: data rate is 1/2 of the tx input high requency clock 2'b10: data rate is the same of the tx input high requency clock 2'b11: data rate is 2 times of the tx input high requency clock
P_TX_BUSWIDTH	I	3	TX buswidth control; 000:8bit 001: 10bit 010:16bit 011: 20bit 100: 32bit 101: 40bit others:not allowed

Port Name	Direction	Data width	Functional description
P_TX_FREERUN_BUSWIDTH	I	3	TX freerun buswidth control; 3'b000: 8bit; 3'b001: 10bit; 3'b010: 16bit; 3'b011: 20bit; 3'b100: 32bit; 3'b101: 40bit; others: reserved.
P_TX_MARGIN	I	3	for pcie mode select transmitter voltage level, from core
P_TX_PMA_RST	I	1	tx module reset, high active
P_TX_LANE_POWERDOWN	I	1	tx_lane_powerdown
P_TX_PIC_EN	I	1	tx pi controller enable from fabric
P_RX_RATE	I	2	Rx rate control signal; 2'b00: data rate is 1/4 of the rx input high requency clock 2'b01: data rate is 1/2 of the rx input high requency clock 2'b10: data rate is the same of the rx input high requency clock 2'b11: data rate is 2 times of the rx input high requency clock
P_RX_BUSWIDTH	I	3	Rx buswidth control; 000:8bit 001:10bit 010:16bit 011:20bit 100:32bit 101:40bit others:not allowed
P_RX_HIGHZ	I	1	Rx input impedance control 1: Highz 0: Normal
P_CIM_CLK_ALIGNER_RX	I	8	The CLK Aligner delay step selection on the receiving side is an asynchronous signal.
P_CIM_CLK_ALIGNER_TX	I	8	The CLK Aligner delay step selection on the sending side is an asynchronous signal
P_ALIGN_MODE_VALID_RX	I	1	clk aligner rx align mode valid
P_ALIGN_MODE_RX	I	2	clk aligner rx align mode
P_ALIGN_MODE_VALID_TX	I	1	clk aligner tx align mode valid
P_ALIGN_MODE_TX	I	3	clk aligner tx align mode
PMA_HPLL_CK0	I	1	Input high-speed clock phase 0, derived from HPLL output
PMA_HPLL_CK90	I	1	Input high speed clock phase 90, derived from HPLL output
PMA_HPLL_CK180	I	1	Input high speed clock phase 180, derived from HPLL output
PMA_HPLL_CK270	I	1	Input high speed clock phase 270, derived from HPLL output
PMA_HPLL_READY_IN	I	1	Lock signal input for HPLL

Port Name	Direction	Data width	Functional description
PMA_HPLL_REFCLK_IN	I	1	Reference clock Input of HPLL
P_LPLL_REFCLK_IN	I	1	Reference clock input for LPLL
PMA_TX_SYNC_HPLL_IN	I	1	Tx sync signal (synchronized to HPLL clock domain)
P_TX_RATE_CHANGE_ON_0	I	1	global tx rate change enable0 input
P_TX_RATE_CHANGE_ON_1	I	1	global tx rate change enable1 input
P_TX_SYNC	I	1	tx_sync signal input port, when the user needs to use the asynchronous tx_sync signal, the sync signal is directly input through this port; when the user needs to use the synchronous tx_sync signal, this port is connected to the TX_SYNC_REFSYNC_O port output by hpll
P_RX_SDN	I	1	Receiver inn
P_RX_SDP	I	1	Receiver inp

Table 349 Bus LANE_COUT_BUS_FORWARD description of each segment

Bit segment	Original corresponding signal	Illustrate
[24]	PCS_TCLK_EN_COUT	pcs_tclk_en output cascade signal
[23]	GEAR_TCLK_EN_COUT	gear_tclk_en output cascade signal
[22]	TFIFO_WR_EN_COUT	tfifo_wr_en output cascade signal
[21]	TFIFO_RD_EN_COUT	tfifo_rd_en output cascade signal
[20]	BRIDGE_RCLK_EN_COUT	bridge_rclk enable cascade signal, bridge_rclk clock domain
[19]	GEAR_RCLK_EN_COUT	gear_rclk_en cascade signal output, gear_rclk clock domain
[18]	RFIFO_RD_EN_COUT	rfifo_rd_en cascade output, bridge_rclk clock domain
[17]	CTC_RD_FIFO_COUT	Master channel skip-add operation data hold operation signal, transmitted to the next channel
[16:15]	CB_START_GAP_COUT	The initial value calculated by the master channel search_gap, which is transmitted to the next channel
[14]	CB_START_FLAG_COUT	The ceb open signal generated by the master channel is transmitted to the next channel
[13]	CB_FIFO_RDEN_COUT	Master channel ceb read enable signal, transmitted to the next channel
[12]	CB_FIFO_INITIAL_COUT	The ceb restart signal generated by the master channel is transmitted to the next channel
[11:8]	CB_SKIP_DEL_MCB_COUT	Master channel skip-del information, transmitted to the next channel
[7:4]	CB_SKIP_ADD_MCB_COUT	Master channel skip-add instructions, transmitted to the next channel
[3:0]	RD_APATTERN_MATCH_COUT	In the master channel fifo read data, apattern-detect information is transmitted to the next channel

Table 350 Bus LANE_CIN_BUS_FORWARD Description of every bit segment

Bit segment	Original corresponding signal	Illustrate
[24]	PCS_TCLK_EN_CIN	Cascade input for pcs_tclk enable signal
[23]	GEAR_TCLK_EN_CIN	Cascade input for gear_tclk_en
[22]	TFIFO_WR_EN_CIN	Cascade input for tfifo_wr_en
[21]	TFIFO_RD_EN_CIN	Cascade input for tfifo_rd_en
[20]	BRIDGE_RCLK_EN_CIN	Cascade input for bridge_rclk enable signal
[19]	GEAR_RCLK_EN_CIN	Cascade input for gear_rclk_en
[18]	RFIFO_RD_EN_CIN	Cascade input for rfifo_rd_en
[17]	CTC_RD_FIFO_CIN	Data hold operation signal under master channel skip-add operation
[16:15]	CB_START_GAP_CIN	The initial value calculated by the master channel search_gap
[14]	CB_START_FLAG_CIN	The ceb open signal generated by the master channel
[13]	CB_FIFO_RDEN_CIN	master channel ceb read enable signal
[12]	CB_FIFO_INITIAL_CIN	ceb restart signal generated by master channel
[11:8]	CB_SKIP_DEL_MCB_CIN	Master channel skip-del information

Bit segment	Original corresponding signal	Illustrate
[7:4]	CB_SKIP_ADD_MCB_CIN	Instructions for master channel skip-add
[3:0]	RD_APATTERN_MATCH_CIN	Master channel fifo read data, apattern-detect information

4. Parameter description

Table 351 GTP_HSSTHP_LANE parameter list

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
PCS_DYN_DLY_SEL_RX	<string >	"FALSE", "TRUE"	"FALSE"	Delay line default setting selection, "TRUE" is valid, select cim_clk_aligner_tx
PCS_PMA_RCLK_POLINV	<string >	"PMA_RCLK"; "REVERSE_OF_PMA_RCLK"	"PMA_RCLK"	Pma_tclk polarity control; "PMA_RCLK": pma_rclk "REVERSE_OF_PMA_RCLK": reverse of pma_rclk
PCS_PCS_RCLK_SEL	<string >	"RCLK"; "HARDWIRE"; "PMA_RCLK"; "PMA_TCLK"	"RCLK"	pcs_rclk select; "RCLK": rclk; "HARDWIRE": hardwire to 0; "PMA_RCLK": pma_rclk; "PMA_TCLK": pma_tclk
PCS_GEAR_RCLK_SEL	<string >	"RCLK"; "RCLK2"; "PMA_RCLK"; "PMA_TCLK"	"RCLK"	gear_rclk select; "RCLK": rclk; "RCLK2": rclk2; "PMA_RCLK": pma_rclk; "PMA_TCLK": pma_tclk
PCS_RCLK2FABRIC_SEL	<string >	"HARD_1"; "CLOCK_DLY"; "FABRIC_CLK_DIV"; "FABRIC_CLK"	"HARD_1"	rclk2fabric select; "HARD_1": hardwire to 1; "CLOCK_DLY": from clock dly; "FABRIC_CLK_DIV": to_fabric_clk_div; "FABRIC_CLK": to_fabric_clk
PCS_SCAN_INTERVAL_RX	<string >	"4_CLOCKS"; "8_CLOCKS"; "16_CLOCKS"; "32_CLOCKS"	"4_CLOCKS"	San mode inc interval; "4_CLOCKS": 4 clocks "8_CLOCKS": 8 clocks "16_CLOCKS": 16 clocks "32_CLOCKS": 32 clocks
PCS_BRIDGE_RCLK_SEL	<string >	"RCLK"; "RCLK2"	"RCLK"	bridge_rclk select "RCLK": rclk; "RCLK2": rclk2;
PCS_RCLK_POLINV	<string >	"RCLK"; "REVERSE_OF_PMA_RCLK"	"RCLK"	pcs_rclk polarity control; "RCLK": rclk "REVERSE_OF_PMA_RCLK": reverse of pma_rclk
PCS_TO_FABRIC_CLK_SEL	<string >	"PMA_RCLK"; "PMA_TCLK"	"PMA_RCLK"	to select pma_rclk or pma_tclk as rclk2fabric source
PCS_CLK2ALIGNER_SEL	<string >	"TO_FABRIC_CLK"; "TO_FABRIC_CLK_DIV"	"TO_FABRIC_CLK"	to enable clk div2 branch for clk2aligner, "TO_FABRIC_CLK": disable div2 "TO_FABRIC_CLK_DIV": enable div2

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_TO_FABRIC_CLK_DIV_EN	<string >	"FALSE", "TRUE"	"FALSE"	to enable clk div2 function; "FALSE": disable;"TRUE":enable
PCS_AUTO_NEAR_LOOP_EN	<string >	"FALSE", "TRUE"	"FALSE"	to enable pcs_near_loop_en control; "FALSE": disable;"TRUE":enable
PCS_PCS_RCLK_EN	<string >	"FALSE", "TRUE"	"FALSE"	Configured as "TRUE", the valid clock enable signal corresponding to clk will be output, otherwise the corresponding clock enable signal is all 1
PCS_BRIDGE_PCS_RCLK_EN_SEL	<string >	"HARD_1"; "FROM_CLK_DI V"; "FROM_PCS_RC LK_EN"; "TOGGLE"	"HARD_1"	bridge_pcs_rclk_en control: "HARD_1": hardwite to 1; "FROM_CLK_DIV": from clock divider; "FROM_PCS_RCLK_EN": from pcs_rclk_en; "TOGGLE": toggle
PCS_BRIDGE_RCLK_EN_SEL	<string >	"HARD_0"; "TOGGLE"; "HARD_1"	"HARD_0"	bridge_rclk_en control: "HARD_0": hardwire to 0; "TOGGLE": toggle; "HARD_1": hardwire to 1'b1
PCS_GEAR_RCLK_EN_SEL	<string >	"HARD_0"; "FALL_EDGE_SAMPLE_ CLKMODE6"; "FROM_CLK_DIV"; "SYNC_BRIDGE_PCS_ RCLK_EN"; "SYNC_PCS_RCLK_EN"; "SYNC_BRIDGE_RCLK_EN"	"HARD_0"	gear_rclk_en control: "HARD_0": hardwire to 0; "FALL_EDGE_SAMPLE_CLKMOD E6": falling edge sample @rx clkmode6; "FROM_CLK_DIV": from clock divider; "SYNC_BRIDGE_PCS_RCLK_EN": sync with bridge_pcs_rclk_en; "SYNC_PCS_RCLK_EN": sync with pcs_rclk_en "SYNC_BRIDGE_RCLK_EN": sync with bridge_rclk_en
PCS_NEGEDGE_EN_RX	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to enable falling edge alignment
PCS_PCS_RX_RSTN	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": pcs_rx_rstn is released, "FALSE": pcs_rx_rstn is reset
PCS_BRIDGE_PCS_RSTN	<string >	"FALSE", "TRUE"	"FALSE"	External bridge unit rstn from configuration
PCS_TO_FABRIC_RST_EN	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to mask pcs_rx_rstn for clk div2 function
PCS_BYPASS_GEAR_RRSTN	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to bypass gear_rclk sync for following rst gen
PCS_BYPASS_BRIDGE_RRSTN	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to bypass bridge_rclk sync for following rst gen
PCS_ALIGNER_EN_RX	<string >	"FALSE", "TRUE"	"FALSE"	set as " FALSE " to force pi_clk related logic (ssc ctrl & clk aligner) in reset state;
PCS_RX_SLAVE	<string >	"MASTER"; "SLAVE"	"MASTER"	set as "SLAVE" for rx channel clock bonding slave

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_RX_CA	<integer>	0 to 255	0	clk aligner rx setting step from configuration
PCS_SUM_THRESHOLD_RX	<integer>	0 to 31	0	Aligner vote threshold
PCS_AVG_CYCLES_RX	<integer>	0 to 7	0	The number of cycles counted by Aligner vote: 0: 16 clocks; others: cfg_avg_cycles_rx * 2
PCS_REG_PMA_RX2TX_ PLOOP_EN	<string >	"FALSE", "TRUE"	"FALSE"	PMA_RX2TX remote parallel loopback clk, data enable
PCS_REG_PMA_RX2TX_ PLOOP_FIFOEN	<string >	"FALSE", "TRUE"	"FALSE"	PMA_RX2TX remote parallel loopback FIFO enable
PCS_STEP_SIZE_RX	<integer>	0 to 3	0	Aligner filter setting, set the DC gain of the filter 0: gain=2 1: gain =4 2: gain=8 3: gain=16
PCS_REV_CNT_LIMIT_RX	<integer>	0 to 3	0	Aligner filter setting, toggle threshold for noise detection of filter 2 0: rollover threshold is 0 1: flipping threshold is 1 2: flipping threshold is 3 3: The rollover threshold is 7
PCS_FILTER_CNT_SIZE_ RX	<integer>	0 to 3	0	Aligner filter setting, calculation time control threshold for filter 2 0: the duration is 31 1: Duration is 63 2: The duration is 127 3: The duration is 255
PCS_DLY_ADJUST_ SIZE_RX_3_0	<integer>	0 to 15	0	Delay line adjust step setting at monitor mode
PCS_DLY_REC_SIZE_RX	<integer>	0 to 15	0	Delay line adjust step setting at recovery mode
PCS_ALIGN_THRD_RX	<integer>	0 to 63	0	Monitor mode judges the threshold of align
PCS_DLY_ADJUST_ SIZE_RX_4	<integer>	0, 1	0	Delay line adjust step setting at monitor mode
PCS_CFG_DEC_TYPE_EN	<string >	"FALSE", "TRUE"	"FALSE"	Rx direction pcie mode switching enable port
PCS_RXBRIDGE_GEAR_SEL	<string >	"FALSE", "TRUE"	"FALSE"	Configure the order of selecting the bridge unit module and the gear module in the rx direction "FALSE": the bridge unit comes first; "TRUE": the gear module comes first
PCS_GE_AUTO_EN	<string >	"FALSE", "TRUE"	"FALSE"	ge auto enable signal in CTC, "TRUE" is valid, detect C and replace C to IDLE.
PCS_RXBRG_FULL_CHK_EN	<string >	"FALSE", "TRUE"	"FALSE"	rx bridge unit fifo full detection enable. "FALSE": disable; "TRUE": enable.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_RXBRG_EMPTY_CHK_EN	<string >	"FALSE", "TRUE"	"FALSE"	rx bridge unit fifo read empty detection enable. "FALSE": disable; "TRUE": enable.
PCS_IFG_EN	<string >	"FALSE", "TRUE"	"FALSE"	rx bridge unit fifo read empty detection enable. "FALSE": disable; "TRUE": enable.
PCS_FLP_FULL_CHK_EN	<string >	"FALSE", "TRUE"	"FALSE"	Far loop fifo full detection enable. "FALSE": disable; "TRUE": enable.
PCS_FLP_EMPTY_CHK_EN	<string >	"FALSE", "TRUE"	"FALSE"	Far loop fifo read empty detection enable. "FALSE": disable; "TRUE": enable.
PCS_RX_POLARITY_INV	<string >	"DELAY"; "BIT_POLARITY_ INVERSION"; "BIT_REVERSAL"; "BIT_POLARITY &REVERSAL"	"DELAY"	"DELAY": delay; "BIT_POLARITY_INVERSION": bit polarity invers-ion; "BIT_REVERSAL": bit reversal; "BIT_POLARITY&REVERSAL": polarity inversion and bit reversal.
PCS_FARLP_PWR_ REDUCTION	<string >	"FALSE", "TRUE"	"FALSE"	Rx far loop power reduction option "TRUE": power reduction; "FALSE": normal
PCS_RXPRBS_PWR_ REDUCTION	<string >	"NORMAL"; "POWER_REDUCTION";	"NORMAL"	Rx prbs power reduction option "TRUE": power reduction; "FALSE": normal
PCS_WDALIGN_PWR_ REDUCTION	<string >	"NORMAL"; "POWER_REDUCTION";	"NORMAL"	Rx wordalign power reduction option "TRUE": power reduction; "FALSE": normal
PCS_RXDEC_PWR_ REDUCTION	<string >	"NORMAL"; "POWER_REDUCTION";	"NORMAL"	Rx 8b10b decoder power reduction option "TRUE": power reduction; "FALSE": normal
PCS_RXBRG_PWR_ REDUCTION	<string >	"NORMAL"; "POWER_REDUCTION";	"NORMAL"	Rx bridge unit power reduction option "TRUE": power reduction; "FALSE": normal
PCS_RXTEST_PWR_ REDUCTION	<string >	"NORMAL"; "POWER_REDUCTION";	"NORMAL"	Rx teststatus power reduction option "TRUE": power reduction; "FALSE": normal
PCS_WA_SOS_DET_TOL	<integer>	0 to 15	0	cfg_wa_sos_det_tolerance, sos detect fault tolerance settings
PCS_WA_SE_DET_TOL	<integer>	0 to 15	0	cfg_wa_skip_end_det_tolerance, Skip end detect fault tolerance setting
PCS_RX_SAMPLE_UNION	<string >	"FALSE", "TRUE"	"FALSE"	Enable input bit width 10bit to synthesize 20bit for 8b10b_8bit mode
PCS_NEAR_LOOP	<string >	"FALSE", "TRUE"	"FALSE"	near-end loop back data choose signal. "FALSE": disable; "TRUE": enable.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_BYPASS_WORD_ALIGN	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": means the module is bypassed; "FALSE": Indicates that the module is valid
PCS_BYPASS_DENC	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": means the module is bypassed; "FALSE": Indicates that the module is valid
PCS_RX_ERRCNT_CLR	<string >	"FALSE", "TRUE"	"FALSE"	prbs gen mode, insert error
PCS_RX_CODE_MODE	<string >	"DUAL_8B10B"; "QUAL_8B10B"; "16BIT_64B66B"; "16BIT_64B67B"; "32BIT_64B66B"; "32BIT_64B67B"; "32BIT_CAUI_64 B66B"; "32BT_128B130B"	"DUAL_8B10B"	set RX decoder mode; "DUAL_8B10B": dual_8b10b; "QUAL_8B10B": quad_8b10b; "16BIT_64B66B": 16bit_64b66b; "16BIT_64B67B": 16bit_64b67b; "32BIT_64B66B": 32bit_64b66b; "32BIT_64B67B": 32bit_64b67b; "32BIT_CAUI_64B66B": 32bit_caui_64b66b; "32BT_128B130B": 32bit_128b130b;
PCS_RX_BYPASS_GEAR	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": means the module is bypassed; "FALSE": Indicates that the module is valid
PCS_ERRDETECT_SILENCE	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": k-code decoding error will be overwritten forcibly, no error will be reported; "FALSE": An error will be reported if there is an error in k-code decoding; for k/d code fb/fd of 8b10b
PCS_RX_DATA_MODE	<string >	"8BIT"; "10BIT"; "16BIT"; "20BIT"; "32BIT"; "40BIT"	"8BIT"	rx internal datawidth; "8BIT": 8bit; "10BIT": 10bit; "16BIT": 16bit; "20BIT": 20bit; "32BIT": 32bit; "40BIT": 40bit; others: reserved
PCS_CA_DYN_CLY_EN_RX	<string >	"FALSE", "TRUE"	"FALSE"	The enable signal output by clk aligner rx to the external delay chain
PCS_CFG_APATTERN_ STATUS_DELAY	<string >	"NO_DELAY" "DELAY_ONE_CYCLE"; "DELAY_TWO_CYCLE"; "DELAY_THREE_CYCLE"	"DELAY_ ONE_CYCLE"	In channel bonding mode and ctc&&channel bonding mode, the delay coefficient of the salve channel output data apattern detection result. "NO_DELAY": No delay; "DELAY_ONE_CYCLE": Delay the apattern_status of this salve channel by one cycle, cascade transmission to the next channel; "DELAY_TWO_CYCLE": Delay the apattern_status of this salve channel by two cycles, cascade transmission to the next channel ; "DELAY_THREE_CYCLE": The apattern_status of this salve channel is delayed by three cycles, cascaded and transmitted to the next channel.
PCS_RX_PRBS_MODE	<string >	"DISABLE"; "PRBS_7"; "PRBS_15"; "PRBS_23"; "PRBS_31"	"DISABLE"	Receiver side PRBS Checker mode selection "DISABLE": Disable PRBS checker; "PRBS_7": PRBS7 mode; "PRBS_15": PRBS15 mode; "PRBS_23": PRBS23 mode; "PRBS_31": PRBS31 mode.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_ALIGN_MODE	<string >	"1GB"; "10GB"; "RAPIDIO"; "OUTSIDE_LSM"	"1GB"	choose link state machine. "1GB": 1gb Ethernet lsm; "10GB": 10gb Ethernet lsm; "RAPIDIO": RapidIO interconnect; "OUTSIDE_LSM": outside lsm.
PCS_COMMA_DET_MODE	<string >	"PATTERN_DETECT"; "PCS_BIT_SLIP_RISE"; "PMA_RX_CLK_SLIP"; "PCS_BIT_SLIP_BOTH"	"PATTERN_ DETECT"	Word alignment mode selection: "PATTERN_DETECT": pattern detect; "PCS_BIT_SLIP_RISE ": pcs bit slip(slip sig only posedge); "PMA_RX_CLK_SLIP": pma rx_clk_slip; "PCS_BIT_SLIP_BOTH": pcs bit slip(slip sig both edge).
PCS_RAPID_VMIN_1	<integer>	0 to 255	0	least 8 bits of vmin, for lsm of rapid
PCS_RAPID_VMIN_2	<integer>	0 to 255	0	most 8 bits of vmin, for lsm of rapid
PCS_RXBU_WIDER_EN	<string >	"80BIT"; "40/20BIT"	"40/20BIT"	"80BIT": datawidth is 80bit, ; "40/20BIT": datawidth is 40/20 bit
PCS_RAPID_IMAX	<integer>	0 to 7	0	IMAX for lsm of rapid, Rapid IO Link State Machine lock state detection bytes
PCS_RX_SPLIT	<string >	"SPLIT_22BIT_11 BIT"; "UNION_22BIT_4 4BIT"; "UNION_44BIT_8 8BIT"	"SPLIT_ 22BIT_11BIT"	Configure the split/union mode of rx gearbox:" SPLIT_22BIT_11BIT": split 22bit to 11bit, for 8b10b_8bit; "UNION_22BIT_44BIT": union 22bit to 44bit; "UNION_44BIT_88BIT": union 44bit to 88bit; others: reserved
PCS_RXBRG_END_ PACKET_9_8	<integer>	0 to 3	0	end symbol of packet before the inter frame gap
PCS_RXBRG_END_ PACKET_7_0	<integer>	0 to 255	0	end symbol of packet before the inter frame gap
PCS_CTC_MAX_DEL	<integer>	0 to 3	0	Set the number of deletions when ctc is deleted continuously
PCS_COMMA_REG0_9_8	<integer>	0 to 3	0	the last 2 bits of comma_reg0
PCS_COMMA_REG1_9_8	<integer>	0 to 3	0	the last 2 bits of comma_reg1
PCS_COMMA_MASK_9_8	<integer>	0 to 3	0	the last 2 bits of comma_mask
PCS_COMMA_REG0_7_0	<integer>	0 to 255	0	the first 8 bits of comma_reg0
PCS_COMMA_REG1_7_0	<integer>	0 to 255	0	the first 8 bits of comma_reg0
PCS_COMMA_MASK_7_0	<integer>	0 to 255	0	the first 8 bits of comma_mask
PCS_FLP_WRADDR_START	<integer>	0 to 15	0	Set farloop's fifo initial write address

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_FLP_RDADDR_START	<integer>	0 to 15	0	Set farloop's fifo initial read address
PCS_CFG_RX_BRIDGE_ CLK_POLINV	<string >	"FALSE", "TRUE"	"FALSE"	rx_bridge_clk polarity control in rx bridge unit; "FALSE":rx_bridge_clk "TRUE":reverse of rx_bridge_clk
PCS_CTC_MODE_RD_SE L	<string >	"NOMINAL_EMPTY"; "NOMINAL_HALF_ FULL_MODE"	"NOMINAL_ EMPTY"	When the channel selects the two working modes of ctc mode and ctc&&channel bonding mode, configure the ctc working mode of the read clock domain. "NOMINAL_HALF_FULL_MODE": nominal half full mode; "NOMINAL_EMPTY": nominal empty mode.
PCS_CTC_AFULL	<integer>	0 to 63	0	Set CTC async FIFO almost full threshold
PCS_FAST_LOCK_GEAR_EN	<string >	"FALSE", "TRUE"	"FALSE"	Enable fast lock gear in rx bridge unit
PCS_CTC_MODE_WR_SE L	<string >	"NOMINAL_EMPTY"; "NOMINAL_HALF_ FULL_MODE"	"NOMINAL_ EMPTY"	When the channel selects the two working modes of ctc mode and ctc&&channel bonding mode, configure the ctc working mode of the write clock domain. "NOMINAL_HALF_FULL_MODE": nominal half full mode; "NOMINAL_EMPTY": nominal empty mode.
PCS_CTC_AEMPTY	<integer>	0 to 63	0	Configure the almost empty waterline (threshold) of the CTC asynchronous FIFO
PCS_CTC_MODE	<string >	"ONE_BYTE"; "GE_TWO_BYTE "; "RXAUI_TWO_BYTE"; "PCIE_ONE_BYTE"; "PCIE_THREE_BYTE"; "PCIE_FOUR_BYTE"; "PCIE_GEN3"	"ONE_BYTE "	CTC mode selection "ONE_BYTE": one_byte (related with cfg_ctc_max_del); "GE_TWO_BYTE": ge_two_byte; "RXAUI_TWO_BYTE": rxaui_two_byte; "PCIE_ONE_BYTE": pcie_one_byte; "PCIE_THREE_BYTE": pcie_three_byte; "PCIE_FOUR_BYTE": pcie_four_byte; "PCIE_GEN3": pcie_gen3.
PCS_RXBRIDGE_MODE	<string >	"BYPASS"; "BRIDGE_FIFO_MODE"; "CTC_MODE"; "CHANNEL_ BONDING_MODE"; "CTC&CHANNEL_ BONDING_MODE"	"BYPASS"	Channel working mode selection: "BYPASS": bypass mode; "BRIDGE_FIFO_MODE":bridge fifo mode; "CTC_MODE":ctc mode; "CHANNEL_BONDING_MODE": channel bonding mode; "CTC&CHANNEL_BONDING_MODE": ctc&&channel bonding mode.
PCS_CTC_ADD_MAX	<integer>	0 to 3	0	skip insert max number

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_CFG_PHDET_EN_RX	<string >	"FALSE", "TRUE"	"FALSE"	The enable signal output to the phase detect module in the rx direction
PCS_WA_SDS_DET_TOL	<integer>	0 to 15	0	Sds detect fault tolerance settings
PCS_CEB_MODE	<string >	"10GB"; "RAPIDIO"; "OUTSIDE_LSM"	"10GB"	State machine selection for channel bonding "10GB": 10gb Ethernet lsm "RAPIDIO": RapidIO interconnect "OUTSIDE_LSM": external state machine others: reserved
PCS_APATTERN_MODE	<string >	"ONE_BYTE"; "TWO_BYTE"	"ONE_BYTE "	In channel bonding mode, mode selection: "ONE_BYTE": one byte mode, at this time the Apattern in each channel is single bit; "TWO_BYTE": two byte mode, at this time, the Apattern to be parsed in each channel is two consecutive bits
PCS_A_REG0_8	<string >	"FALSE", "TRUE"	"FALSE"	apatern(the last 1 bit)
PCS_RXBRG_WADDR_START	<integer>	0 to 127	0	Configure the initial address of the built-in FIFO write address
PCS_A_REG1_8	<string >	"FALSE", "TRUE"	"FALSE"	apatern(the last 1 bit)
PCS_RXBRG_RADDR_START	<integer>	0 to 127	0	Configure the initial address of the built-in FIFO read address
PCS_A_REG0_7_0	<integer>	0 to 255	0	apatern(the first 8 bit)
PCS_A_REG1_7_0	<integer>	0 to 255	0	apatern(the first 8 bit)
PCS_CEB_RAPIDLS_MMAX	<integer>	0 to 7	0	When the built-in state machine is configured as rapid_IO, configure the number of bonding channels through cfg_ceb_rapidlsm_mmax
PCS_CEB_DETECT_TIME	<integer>	0 to 31	0	Participate in the generation of the initialization signal fifo_initial
PCS_WL_FIFO_RD	<integer>	0 to 63	0	After the internal FIFO is enabled and the data water level reaches cfg_wl_fifo_rd, data reading starts
PCS_SKIP_REG0_9_8	<integer>	0 to 3	0	It only occupies 2 bits, bit0 and the next parameter together form a 9-bit skip pattern, bit1 is used as word align lsm output enable control, high effective.
PCS_SKIP_REG0_7_0	<integer>	0 to 255	0	the first 8 bit of skip pattern of ctc reg0 low 8bit)
PCS_CFG_CONTI_SKP_SET	<integer>	0, 1	0	Continue skip del enable
PCS_CFG_RX_BASE_ ADV_MODE	<string >	"BASE_MODE"; "ADV_MODE"	"BASE_MODE"	rx direction base mode and adv mode setting: "BASE_MODE": base; "ADV_MODE": adv
PCS_SKIP_REG1_9_8	<integer>	0 to 3	0	the last 2 bit of skip pattern of ctc reg1
PCS_SKIP_REG2_9_8	<integer>	0 to 3	0	the last 2 bit of skip pattern of ctc reg2
PCS_SKIP_REG3_9_8	<integer>	0 to 3	0	the last 2 bit of skip pattern of ctc reg3
PCS_SKIP_REG1_7_0	<integer>	0 to 255	0	the first 8 bit of skip pattern of ctc reg1
PCS_SKIP_REG2_7_0	<integer>	0 to 255	0	the first 8 bit of skip pattern of ctc reg2
PCS_SKIP_REG3_7_0	<integer>	0 to 255	0	the first 8 bit of skip pattern of ctc reg3
PCS_CFG_PRBS_ERR_O_ SEL	<integer>	0, 1	0	prbs error output: 0: pcs prbs error; 1: pma prbs error

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_CFG_PD_DELAY_RX	<integer>	0 to 3	0	Set rx phase detect module precision: 0: 100ps; 1: 150ps; 2: 200ps; 3: 300ps
PCS_WR_START_GAP	<integer>	0 to 31	0	Channel delay configuration in beats, using typical configuration defaults
PCS_MIN_IFG	<integer>	0 to 31	0	Configure the minimum inter frame gap between data
PCS_INT_RX_MASK_0	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to mask int triggered by rx_bridge_unit async fifo underflow
PCS_INT_RX_MASK_1	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to mask int triggered by rx_bridge_unit async fifo underflow
PCS_INT_RX_MASK_2	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to mask int triggered by rx_bridge_unit async fifo underflow
PCS_INT_RX_MASK_3	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to mask int triggered by rx_bridge_unit async fifo underflow
PCS_INT_RX_MASK_4	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to mask int triggered by rx_bridge_unit async fifo underflow
PCS_INT_RX_MASK_5	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to mask int triggered by rx_bridge_unit async fifo underflow
PCS_INT_RX_CLR_5	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to clr int triggered by pcs13g_rx_bridge_unit async fifo underflow,
PCS_INT_RX_CLR_4	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to clr int triggered by pcs13g_rx_bridge_unit async fifo overflow
PCS_INT_RX_CLR_3	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to clr int triggered by pcs13g_rx_bridge_unit async fifo almost empty
PCS_INT_RX_CLR_2	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to clr int triggered by pcs13g_rx_bridge_unit async fifo almost full
PCS_INT_RX_CLR_1	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to clr int triggered by pcs13g_rx_bridge_unit mcb_status
PCS_INT_RX_CLR_0	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to clr int triggered by pcs13g_wa_top lsm loss of sync
PCS_EM_CNT_RD_EN	<string >	"FALSE", "TRUE"	"FALSE"	enable signal for locking the data of error and sample counts during the accumulating process
PCS_EM_CTRL_SEL	<string >	"SIGNAL_CTRL"; "REGISTERS_CTRL"	"SIGNAL_CTRL "	config signal for mode selection from config registers or mode control signal; "SIGNAL_CTRL": mode control corresponding to em_mode_ctrl(port); "REGISTERS_CTRL": mode control corresponding to cfg_em_mode_ctrl(register).

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_EM_MODE_CTRL	<string >	"ERROR_COUNTER"; "CAPTURE_DATA"; "HOLD"	"HOLD"	config signal for executing mode: "ERROR_COUNTER": run error couner mode "CAPTURE_DATA": run diagnostic mode, CAPTURE DATA mode; "HOLD": hold mode; others: Not allowed.
PCS_EM_RD_CONDITION	<string >	"CHECK_ERROR"; "CHECK_SAMPLE"; "TRIGGER"; "DIRECT"	"TRIGGER"	config signal for read condition during CAPTURE DATA mode
PCS_EM_SP_PATTERN_7_0	<integer>	0 to 255	0	qualifing BER measurements, Seting for searching for a special pattern of RDATA
PCS_EM_SP_PATTERN_15_8	<integer>	0 to 255	0	qualifing BER measurements, Seting for searching for a special pattern of RDATA
PCS_EM_SP_PATTERN_23_16	<integer>	0 to 255	0	qualifing BER measurements, Seting for searching for a special pattern of RDATA
PCS_EM_SP_PATTERN_31_24	<integer>	0 to 255	0	qualifing BER measurements, Seting for searching for a special pattern of RDATA
PCS_EM_SP_PATTERN_39_32	<integer>	0 to 255	0	qualifing BER measurements, Seting for searching for a special pattern of RDATA
PCS_EM_SP_PATTERN_47_40	<integer>	0 to 255	0	qualifing BER measurements, Seting for searching for a special pattern of RDATA
PCS_EM_SP_PATTERN_55_48	<integer>	0 to 255	0	qualifing BER measurements, Seting for searching for a special pattern of RDATA
PCS_EM_SP_PATTERN_63_56	<integer>	0 to 255	0	qualifing BER measurements, Seting for searching for a special pattern of RDATA
PCS_EM_SP_PATTERN_71_64	<integer>	0 to 255	0	qualifing BER measurements, Seting for searching for a special pattern of RDATA
PCS_EM_SP_PATTERN_79_72	<integer>	0 to 255	0	qualifing BER measurements, Seting for searching for a special pattern of RDATA
PCS_EM_PMA_MASK_7_0	<integer>	0 to 255	0	setting for masking those bits not included in qualifier pattern

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_EM_PMA_MASK_15_8	<integer>	0 to 255	0	setting for masking those bits not included in qualifier pattern
PCS_EM_PMA_MASK_23_16	<integer>	0 to 255	0	setting for masking those bits not included in qualifier pattern
PCS_EM_PMA_MASK_31_24	<integer>	0 to 255	0	setting for masking those bits not included in qualifier pattern
PCS_EM_PMA_MASK_39_32	<integer>	0 to 255	0	setting for masking those bits not included in qualifier pattern
PCS_EM_PMA_MASK_47_40	<integer>	0 to 255	0	setting for masking those bits not included in qualifier pattern
PCS_EM_PMA_MASK_55_48	<integer>	0 to 255	0	setting for masking those bits not included in qualifier pattern
PCS_EM_PMA_MASK_63_56	<integer>	0 to 255	0	setting for masking those bits not included in qualifier pattern
PCS_EM_PMA_MASK_71_64	<integer>	0 to 255	0	setting for masking those bits not included in qualifier pattern
PCS_EM_PMA_MASK_79_72	<integer>	0 to 255	0	setting for masking those bits not included in qualifier pattern
PCS_EM_EYED_MASK_7_0	<integer>	0 to 255	0	setting for masking those bits not used in SDATA
PCS_EM_EYED_MASK_15_8	<integer>	0 to 255	0	setting for masking those bits not used in SDATA
PCS_EM_EYED_MASK_23_16	<integer>	0 to 255	0	setting for masking those bits not used in SDATA
PCS_EM_EYED_MASK_31_24	<integer>	0 to 255	0	setting for masking those bits not used in SDATA
PCS_EM_EYED_MASK_39_32	<integer>	0 to 255	0	setting for masking those bits not used in SDATA
PCS_EM_EYED_MASK_47_40	<integer>	0 to 255	0	setting for masking those bits not used in SDATA
PCS_EM_EYED_MASK_55_48	<integer>	0 to 255	0	setting for masking those bits not used in SDATA
PCS_EM_EYED_MASK_63_56	<integer>	0 to 255	0	setting for masking those bits not used in SDATA
PCS_EM_EYED_MASK_71_64	<integer>	0 to 255	0	setting for masking those bits not used in SDATA
PCS_EM_EYED_MASK_79_72	<integer>	0 to 255	0	setting for masking those bits not used in SDATA
PCS_EM_PRESCALE	<integer>	0 to 31	0	config signal for controlling the pre-scaling of the sample count
PCS_CFG_TEST_STATUS_SEL	<string >	"SEL_PMA_TEST_ STATUS_INT"; "SEL_CA_TEST_ STATUS_RX_19_0"; "SEL_CA_TEST_ STATUS_RX_39_ 20"; "SEL_CA_TEST_ STATUS_TX_19_ 0"; "SEL_CA_TEST_ STATUS_TX_39_20"	"SEL_PMA_TEST_ STATUS_INT"	Options for test status

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_CFG_DIFF_CNT_BND_RX	<integer>	0 to 15	0	Noise detection threshold of filter 1, 4bit unsigned number
PCS_CFG_FLT_SEL_RX	<string >	"FALSE", "TRUE"	"FALSE"	Filter Selection Control "FALSE": The input signal is processed with filter 1 "TRUE": Apply filter 2 to the input signal to process
PCS_FILTER_BND_RX	<integer>	0 to 255	0	Set the accumulation threshold of filter 1, 8bit unsigned number. Minimum value 1, maximum value 255.
PCS_TCLK2FABRIC_DIV_ RST_M	<string >	"FALSE", "TRUE"	"FALSE"	Set as "TRUE" to mask reset function for clock divider
PCS_TX_PMA_TCLK_POLINV	<string >	"PMA_TCLK"; "REVERSE_OF_ PMA_TCLK"	"PMA_TCLK "	Pma_tclk polarity control; "PMA_TCLK": pma_tclk; "REVERSE_OF_PMA_TCLK": reverse of pma_tclk
PCS_TX_TCLK_POLINV	<string >	"TCLK"; "REVERSE_OF_TCLK"	"TCLK"	Tclk polarity control; "TCLK": tclk; "REVERSE_OF_TCLK": reverse of tclk
PCS_PCS_TCLK_SEL	<string >	"PMA_TCLK"; "TCLK"	"PMA_TCLK "	Pcs_tclk select "PMA_TCLK": pma_tclk; "TCLK": tclk
PCS_GEAR_TCLK_SEL	<string >	"PMA_TCLK"; "TCLK2"	"PMA_TCLK "	Gear_tclk select "PMA_TCLK": pma_tclk; "TCLK2": tclk2
PCS_TX_BRIDGE_TCLK_SEL	<string >	"TCLK"; "TCLK2"	"TCLK"	Bridge tclk select "TCLK": tclk; "TCLK2": tclk2
PCS_TCLK2ALIGNER_SEL	<string >	"PMA_TCLK"; "PMA_TCLK_DIV2"; "PMA_FREE_TCLK"	"PMA_TCLK "	clk2aligner select: "PMA_TCLK": pma_tclk; "PMA_TCLK_DIV2": pma_tclk div2; "PMA_FREE_TCLK": pma_free_tclk; others: reserved.
CA_DYN_DLY_EN_TX	<string >	"FALSE", "TRUE"	"FALSE"	The enable signal output by clk aligner tx to the external delay chain
PCS_TX_PCS_CLK_EN_SEL	<string >	"HARDWIRED1"; "CLK_ALIGNER"; "TOGGLE"	"HARDWIRED1"	Pcs_clk_en select; "HARDWIRED1": hardwired to 1; "CLK_ALIGNER": align to clk aligner; "TOGGLE": toggle.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_TX_GEAR_CLK_ EN_SEL	<string >	"HARDWIRED0"; "CLKDIV"; "FALL_EDGE_SAMPLE_ CLKMODE5"; "ALIGN_PCS_CLK_EN"	"HARDWIRED0"	Gear_tclk_en select; "HARDWIRED0": hardwired to 0 "CLKDIV": from clock divider "FALL_EDGE_SAMPLE_CLKMOD E5": falling edge sampe, @tx clk mode 5 "ALIGN_PCS_CLK_EN": align with pcs_clk_en
PCS_TCLK2FABRIC_ DIV_EN	<string >	"FALSE", "TRUE"	"FALSE"	Set as "TRUE" enable clk div2 function
PCS_TCLK2FABRIC_SEL	<string >	"CLK2ALIGNER_N_DIV2"; "CLKDLY_N_DIV2"; "CLK2ALIGNER_DIV2"; "CLKDLY_DIV2"	"CLK2ALIGNER_ N_DIV2"	tclk2fabric select "CLK2ALIGNER_N_DIV2": from clk2aligner, without div2; "CLKDLY_N_DIV2": from clock dly,without div2; "CLK2ALIGNER_DIV2": from clk2aligner, div2; "CLKDLY_DIV2": from clock dly, div2;
DLY_ADJUST_SIZE_TX	<integer>	0 to 31	0	Delay line adjust step setting at monitor mode
PCS_TX_PCS_TX_RSTN	<string >	"FALSE", "TRUE"	"FALSE"	Pcs tx lane rstn from configuration
PCS_TX_CA_RSTN	<string >	"FALSE", "TRUE"	"FALSE"	Rst control for clk aligner
PCS_TX_SLAVE	<string >	"MASTER"; "SLAVE"	"MASTER"	set as "SLAVE" for tx/rx channel clock bonding slave
PCS_TX_CA	<integer>	0 to 255	0	clk aligner rx setting step from configuration
PCS_CFG_PI_CLK_SEL	<integer>	0 to 3	0	1 select pma_tclk_mux; 0 select cfg_clk
PCS_CFG_PI_CLK_ EN_SEL	<string >	"CLK_EN_ROLL"; "CLK_EN_ALWAYS1"	"CLK_EN_ ALWAYS1"	"CLK_EN_ROLL": clk_en roll; "CLK_EN_ALWAYS1": clk en always 1
PCS_CFG_PI_STEP_ SIZE_TX	<integer>	0 to 31	0	Delay line adjust step setting at recovery mode
PCS_CFG_SUM_ THRESHOLD_TX	<integer>	0 to 31	0	Aligner vote threshold
PCS_CFG_AVG_ CYCLES_TX	<integer>	0 to 7	0	The number of cycles counted by Aligner vot: 0: 16 clocks; others: cfg_avg_cycles_tx * 2
PCS_CFG_NEGEDGE_ EN_TX	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to enable falling edge alignment
PCS_CFG_ALIGN_ THRD_TX	<integer>	0 to 127	0	Monitor mode judges the threshold of align
PCS_CFG_SCAN_ INTERVAL_TX	<integer>	0 to 3	0	San mode inc interval: 0: 4 clocks 1: 8 clocks 2: 16 clocks 3: 32 clocks

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_CFG_STEP_SIZE_TX	<integer>	0 to 3	0	Aligner filter setting, set the DC gain of the filter 0: gain=2 1: gain =4 2: gain=8 3: gain=16
PCS_CFG_REV_CNT_ LIMIT_TX	<integer>	0 to 3	0	Aligner filter setting, toggle threshold for noise detection of filter 2 0: rollover threshold is 0 1: flipping threshold is 1 2: flipping threshold is 3 3: The rollover threshold is 7
PCS_CFG_FILTER_CNT_ SIZE_TX	<integer>	0 to 3	0	Aligner filter setting, calculation time control threshold for filter 2 0: the duration is 31 1: Duration is 63 2: The duration is 127 3: The duration is 255
PCS_CFG_PI_DEFAULT_TX	<integer>	0 to 255	0	Pi code default setting
PCS_CFG_PHDET_EN_TX	<string >	"FALSE", "TRUE"	"FALSE"	enable to tx phase detect module
PCS_PMA_TX2RX_PLOOP_EN	<string >	"FALSE", "TRUE"	"FALSE"	PMA_TX2RX_PLOOP_EN near-end parallel loopback enable
PCS_PMA_TX2RX_SLOOP_EN	<string >	"FALSE", "TRUE"	"FALSE"	PMA_TX2RX_SLOOP_EN near-end serial loopback enable
PCS_CFG_DYN_DLY_SEL_TX	<string >	"FALSE", "TRUE"	"FALSE"	Delay line default setting selection, set as "TRUE" selection cim_clk_aligner_tx
PCS_CFG_DLY_REC_SIZE_TX	<integer>	0 to 15	0	Delay line adjust step setting at recovery mode
PCS_TX_DATA_WIDTH_ MODE	<string >	"8BIT"; "10BIT"; "16BIT"; "20BIT"; "32BIT"; "40BIT"	"8BIT"	tx internal datawidth: "8BIT": 8bit; "10BIT": 10bit; "16BIT": 16bit; "20BIT": 20bit; "32BIT": 32bit; "40BIT": 40bit; others: reserved
PCS_TX_BYPASS_ BRIDGE_UINT	<string >	"FALSE", "TRUE"	"FALSE"	""TRUE": means the module is bypassed; "FALSE": Indicates that the module is valid
PCS_TX_BYPASS_ BRIDGE_FIFO	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": means the module is bypassed; "FALSE": Indicates that the module is valid
PCS_TX_BYPASS_GEAR	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": means the module is bypassed; "FALSE": Indicates that the module is valid

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_TX_BYPASS_ENC	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": means the module is bypassed; "FALSE": Indicates that the module is valid
PCS_TX_BYPASS_BIT_SLIP	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": means the module is bypassed; "FALSE": Indicates that the module is valid
PCS_TX_BRIDGE_GEAR_SEL	<string >	"FALSE", "TRUE"	"FALSE"	Configure the order of selecting the bridge unit module and gear module in the tx direction "FALSE": the bridge unit comes first; "TRUE": the gear module comes first
PCS_TXBRG_PWR_ REDUCTION	<string >	"NORMAL"; "POWER_REDUCTION"	"NORMAL"	tx bridge unit power reduction option; "POWER_REDUCTION": power reduction; "NORMAL": normal
PCS_TXGEAR_PWR_ REDUCTION	<string >	"NORMAL"; "POWER_REDUCTION"	"NORMAL"	tx gear power reduction option "POWER_REDUCTION": power reduction; "NORMAL": normal
PCS_TXENC_PWR_ REDUCTION	<string >	"NORMAL"; "POWER_REDUCTION"	"NORMAL"	tx 8b10b encoder power reduction option "POWER_REDUCTION": power reduction; "NORMAL": normal
PCS_TXBSLP_PWR_ REDUCTION	<string >	"NORMAL"; "POWER_REDUCTION"	"NORMAL"	tx bitslip power reduction option "POWER_REDUCTION": power reduction; "NORMAL": normal
PCS_TXPRBS_PWR_ REDUCTION	<string >	"NORMAL"; "POWER_REDUCTION"	"NORMAL"	tx prbs power reduction option "POWER_REDUCTION": power reduction; "NORMAL": normal
PCS_TXBRG_FULL_CHK_EN	<string >	"FALSE", "TRUE"	"FALSE"	tx bridge unit fifo full detection enable. "FALSE": disable, corresponding to Parameter parameter value "FALSE "; "TRUE": enable, corresponding to Parameter parameter value "TRUE ".
PCS_TXBRG_EMPTY_CHK_EN	<string >	"FALSE", "TRUE"	"FALSE"	tx bridge unit fifo empty detection enable. "FALSE": disable, corresponding to Parameter parameter value "FALSE "; "TRUE": enable, corresponding to Parameter parameter value "TRUE ".
PCS_TX_ENCODER_MODE	<string >	"DUAL_8B10B"; "QUAL_8B10B"; "16BIT_64B66B"; "16BIT_64B67B"; "32BIT_64B66B"; "32BIT_64B67B"; "32BIT_CAUI_64B66B"; "32BT_128B130B"; "SINGLE_8B10B"	"DUAL_8B10B"	set TX encoder mode; "DUAL_8B10B": dual_8b10b; "QUAL_8B10B": quad_8b10b; "16BIT_64B66B": 16bit_64b66b; "16BIT_64B67B": 16bit_64b67b; "32BIT_64B66B": 32bit_64b66b; "32BIT_64B67B": 32bit_64b67b; "32BIT_CAUI_64B66B": 32bit_caui_64b66b; "32BT_128B130B": 32bit_128b130b; "SINGLE_8B10B": single_8b10b.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_TX_PRBS_MODE	<string >	"DISABLE"; "PRBS_7"; "PRBS_15"; "PRBS_23"; "PRBS_31"; "LONG1"; "LONG0"; "40UI"; "D10_2"; "PCIE_20"; "PCIE_40"	"DISABLE"	prbs gen mode signal Sending side PRBS generator mode selection "DISABLE": corresponding to Parameter parameter value "DISABLE"; "PRBS_7": corresponds to the Parameter parameter value "PRBS_7"; "PRBS_15": Corresponding to Parameter parameter value "PRBS_15"; "PRBS_23": Corresponding to Parameter parameter value "PRBS_23"; "PRBS_31": Corresponding to Parameter parameter value"PRBS_31; "LONG1": long "1"; "LONG0": long "0"; "40UI": 40UI square wave; "D10_2": D10_2; "PCIE_20": PCIe complianece pattern,for 20bit width; "PCIE_40": PCIe complianece pattern,for 40bit width; Others: reserved.
PCS_TX_DRIVE_REG_MODE	<string >	"NO_CHANGE"; "POLARITY_ REVERSE_EN"; "BIT_REVERSE_EN"; "BOTH_EN"	"NO_CHANGE"	"NO_CHANGE": no change; "POLARITY_REVERSE_EN": enable polarity reverse; "BIT_REVERSE_EN": enable bit reverse; "BOTH_EN": enable both.
PCS_TX_BIT_SLIP_CYCLES	<integer>	0 to 63	0	Determine the number of bits in the Slip in the Tx Bit Slip module
PCS_TX_BASE_ADV_MODE	<string >	"BASE"; "ADV"	"BASE"	base/advance mode cfg: "BASE": 64b66b/67b is base mode; "ADV": 64b66b/67b is advance mode.
PCS_TX_GEAR_SPLIT	<string >	"NO SPILT"; "SPLIT_TX_DATA"	"NO_SPILT"	"SPLIT_TX_DATA": spilt tx data; "NO SPILT": no spilt
PCS_RX_BRIDGE_ CLK_POLINV	<string >	"CLK_INVERT"; "N_CLK_INVERT"	"N_CLK_INVERT"	"CLK_INVERT": clk invert; "N_CLK_INVERT": without clk invert
PCS_PRBS_ERR_LPBK	<string >	"FALSE", "TRUE"	"FALSE"	prbs_err_lpbk control
PCS_TX_INSERT_ER	<string >	"FALSE", "TRUE"	"FALSE"	prbs gen mode, insert error
PCS_ENABLE_PRBS_GEN	<string >	"FALSE", "TRUE"	"FALSE"	enable prbs module
PCS_FAR_LOOP	<string >	"FALSE", "TRUE"	"FALSE"	far-end loop back data choose signal, high valued

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_CFG_ENC_TYPE_EN	<string >	"FALSE", "TRUE"	"FALSE"	tx direction pcie mode switching port enable signal
PCS_TXBRG_WADDR_START	<integer>	0 to 31	0	Configure the initial address of the tx bridge unit built-in FIFO write address
PCS_TXBRG_RADDR_START	<integer>	0 to 31	0	Configure tx bridge unit built-in FIFO write address initial address of
PCS_CFG_TX_PIC_EN	<string >	"DISABLE"; "ENABLE"	"DISABLE"	"ENABLE": enable ssc controller; "DISABLE": disable ssc controller.
PCS_CFG_PIC_DIRECT_INV	<string >	"FALSE", "TRUE"	"FALSE"	invert the cfg_pic_direct; "TRUE":invert;"FALSE": not invert.
PCS_CFG_PI_MOD_CLK_EN	<string >	"FALSE", "TRUE"	"FALSE"	pcs13g_tx_ssc_modulator clock select signal; "FALSE":select txpclk; "TRUE":select txpclk/2.
PCS_CFG_TX_ MODULATOR_OW_EN	<string >	"FALSE", "TRUE"	"FALSE"	tx_ssc_modulator output overwrite signal; "FALSE": no overwrite; "TRUE": overwrite tx_ssc_modulator output with cfg_txpic_ow_val[6:0].
PCS_CFG_TX_PI_SSC_ MODE_EN	<string >	"FALSE", "TRUE"	"FALSE"	ssc mode enable signal, default:1'b1; "FALSE": disable ssc mode; "TRUE": enable ssc mode.
PCS_CFG_TX_PI_OFFSET_ MODE_EN	<string >	"FALSE", "TRUE"	"FALSE"	offset mode enable signal, default:1'b0; "FALSE": disable offset mode; "TRUE": enable offset mode.
PCS_CFG_TX_PI_SSC_ MODE_SEL	<integer>	0 to 3	0	ssc mode selete signal; 0: down-spread; 1: up-spread; 2: center-spread; 3: down-spread;
PCS_CFG_TXDEEMPH_EN	<string >	"FALSE", "TRUE"	"FALSE"	Configure tx_deemp_pre, tx_deemp_main, tx_deemp_post from port or register: "FALSE": from pma_reg_tx register; "TRUE": from fabric port
PCS_PI_STROBE_SEL	<string >	"FALSE", "TRUE"	"FALSE"	Select whether pma_tx_pi_ctrl comes from tx_picode_pic_o or tx_picode_aligner. "FALSE": selection from tx pi ctrl, tx_picode_pic_o; "TRUE": selects from tx clk aligner tx_picode_aligner
PCS_CFG_TX_PIC_GREY_SEL	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": use grey code.
PCS_CFG_PIC_RENEW_INV	<string >	"NORMAL"; "STROB_SIG_INVERT"	"NORMAL"	"STROB_SIG_INVERT": strobe signal invert; "NORMAL": normal.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_CFG_NUM_PIC	<integer>	0 to 7	0	define how many selected clock cycles between generated adjacent pi_strobe
PCS_CFG_TXPIC_OW_EN	<string >	"FALSE", "TRUE"	"FALSE"	"FALSE": tx_picode_ssc_o[7:0], tx_pi_strobe_ssc_o generated by internal logic; "TRUE": generated by cfg
PCS_CFG_TX_SSC_OW_ VALUE_0_7	<integer>	0 to 255	0	overwrite value; If cfg_tx_modulator_ow_en = 1'b1 (true), configurate cfg_txpic_ow_val[6:0] as overwrite value of tx_pi_control_o[6:0] of pcs13g_tx_ssc_modulator, tx_pi_control_o[7] is no use in this case; If cfg_txpic_ow_en = 1'b1 (true), configurate cfg_txpic_ow_val[7:0] as overwrite value of tx_pi_control_o[7:0] of pcs13g_tx_ssc_ctrl; (cfg_tx_modulator_ow_en=1'b1 and cfg_txpic_ow_en=1'b1 at the same time is invalid).
PCS_INT_TX_MASK_0	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to mask int triggered by tx_invalid_k
PCS_INT_TX_MASK_1	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to mask int triggered by tx_invalid_k
PCS_INT_TX_MASK_2	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to mask int triggered by tx_invalid_k
PCS_TX_WPTR_SEL	<string >	"FALSE", "TRUE"	"FALSE"	tx bridge buffer wptr select, "FALSE": wptr; "TRUE": wgraynext
PCS_INT_TX_CLR_2	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to clear int triggered by tx_invalid_k
PCS_INT_TX_CLR_1	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to clear int triggered by tx_bridge_unit async fifo underflow
PCS_INT_TX_CLR_0	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to clear int triggered by tx_bridge_unit async fifo overflow
PCS_CFG_PD_DELAY_TX	<integer>	0 to 3	0	Set phase detect module precision
PCS_CFG_DIFF_CNT_BND_TX	<integer>	0 to 15	0	Noise detection threshold of filter 1, 4bit unsigned number
PCS_CFG_PD_CLK_FR_ CORE_SEL	<string >	"FALSE", "TRUE"	"FALSE"	pd_clk_fr_core select; "FALSE": tclk; "TRUE": tclk2
PCS_CFG_FLT_SEL_TX	<string >	"FALSE", "TRUE"	"FALSE"	Filter Selection Control "FALSE": The input signal is processed with filter 1 "TRUE": The input signal is processed with filter 2

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PCS_FILTER_BND_TX	<integer>	0 to 255	0	Accumulation threshold of filter 1, 8bit unsigned number. Minimum value 1, maximum value 255.
PCS_CFG_TX_SSC_ RANGE_7_0	<integer>	0 to 255	0	set the range of spread spectrum; 10bits unsigned binary code; ssc_range is the range of spread spectrum with unit of ssc.
PCS_CFG_TX_SSC_ MODULATOR_EN	<string >	"FALSE", "TRUE"	"FALSE"	tx_ssc_modulator enable signal, default:1'b1; "FALSE": disable, tx_pi_control_o[6:0]=7'b0; "TRUE": enable.
PCS_CFG_TX_SSC_ SCALE2_SEL	<integer>	0 to 7	0	select the scaling factor2 for different setting of pma_tx_buswidth<2:0>; 0: scaling factor2 = 4; 1: scaling factor2 = 5; 2: scaling factor2 = 8; 3: scaling factor2 = 10; 4: scaling factor2 = 16; 5: scaling factor2 = 20; 6: scaling factor2 = 1; 7: scaling factor2 = 1
PCS_CFG_TX_SSC_ SCALE_SEL	<integer>	0 to 3	0	select the scaling factor for different setting of pma_tx_rate<1:0>; 0: scaling factor = 8; 1: scaling factor = 4; 2: scaling factor = 2; 3: scaling factor = 1
PCS_CFG_TX_SSC_ RANGE_8_9	<integer>	0 to 3	0	set the range of spread spectrum; 10bits unsigned binary code; ssc_range is the range of spread spectrum with unit of ssc.
PCS_CFG_TX_SSC_ MODULATION_STEP_7_0	<integer>	0 to 255	0	set the step of spread spectrum; 9bits unsigned binary code
PCS_CFG_TX_SSC_ OFFSET_7_0	<integer>	0 to 255	0	set the offset of spectrum; 10bit two's complement code, positive value means negative offset, negative value means positive offset
PCS_CFG_TX_SSC_ MODULATION_STEP_8	<integer>	0, 1	0	set the step of spread spectrum; 9bits unsigned binary code
PCS_CFG_TX_SSC_ OFFSET_8_9	<integer>	0 to 3	0	set the offset of spectrum; 10bit two's complement code, positive value means negative offset, negative value means positive offset

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_RX related parameters are as follows:				
PMA_REG_RX_BUSWIDTH	<string >	"20BIT"; "8BIT"; "10BIT"; "16BIT"; "32BIT"; "40BIT"	"40BIT"	Rx buswidth control; "8BIT": 8bit; "10BIT": 10bit; "16BIT": 16bit; "20BIT": 20bit; "32BIT": 32bit; "40BIT": 40bit; others: not allowed
PMA_REG_RX_RATE	<string >	"DIV4"; "DIV2"; "DIV1"; "MUL2"	"DIV4"	Rx rate ctrl "DIV4": data rate is 1/4 of the rx input high requency clock "DIV2": data rate is 1/2 of the rx input high requency clock "DIV1": data rate is the same of the rx input high requency clock "MUL2": data rate is 2 times of the rx input high requency clock
PMA_REG_RX_RATE_EN	<string >	"FALSE", "TRUE"	"FALSE"	When reg_rx_rate_en is "TRUE", ana_rx_rate_o<1:0> selects the reg_rx_rate register; otherwise, selects the pma_rx_rate_i port
PMA_REG_RX_RES_TRIM	<integer>	0 to 63	55	rx impedance matching correction control
PMA_REG_RX_SIGDET_STATUS_EN	<string >	"FALSE", "TRUE"	"FALSE"	When reg_rx_sigdet_status_en is "TRUE", pma_rx_sigdet_status_o is directly controlled by register reg_rx_sigdet_status; rx_sigdet_status_2oob_det_o is directly controlled by register reg_rx_sigdet_status
PMA_REG_CDR_READY_THD_7_0	<integer>	0 to 255	32	When the rms method is used to detect cdr lock, the cdr lock threshold
PMA_REG_CDR_READY_THD_11_8	<integer>	0 to 15	0	When the rms method is used to detect cdr lock, the cdr lock threshold
PMA_REG_RX_BUSWIDTH_EN	<string >	"FALSE", "TRUE"	"FALSE"	When configured to "TRUE", ana_rx_buswidth_o<2:0> from the reg_rx_buswidth register (pma_rx_reg_i[32:30]); otherwise from pma_rx_buswidth_i port
PMA_REG_RX_PCLK_EDGE_SEL	<string >	"POS_EDGE"; "NEG_EDGE"	"POS_EDGE"	Reserved
PMA_REG_CDR_READY_CHECK_CTRL	<integer>	0 to 3	0	cdr ready signal output method selection 0: false lock detection 1: rms method detection, single 2: rms method detection, continuous 3: pi difference method detection
PMA_REG_RX_ICTRL_TRX	<string >	"100PCT"; "87_5PCT"; "112_5PCT"; "125PCT"	"100PCT"	Reserved

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_PRBS_CHK_ WIDTH_SEL	<integer>	0 to 5	1	Selection of the width of Prbs chk 0: 8b 1: 10b 2: 16b 3: 20b 4: 32b 5: 40b
PMA_REG_RX_ICTRL_PIBUF	<string >	"100PCT"; "87_5PCT"; "112_5PCT"; "125PCT"	"100PCT"	Reserved
PMA_REG_RX_ICTRL_PI	<string >	"100PCT"; "87_5PCT"; "112_5PCT"; "125PCT"	"100PCT"	Reserved
PMA_REG_RX_ICTRL_DCC	<string >	"100PCT"; "87_5PCT"; "112_5PCT"; "125PCT"	"100PCT"	Reserved
PMA_REG_TX_RATE	<string >	"DIV1" "DIV4" "DIV2" "MUL2"	"DIV1"	Tx rate ctrl "DIV4": data rate is 1/4 of the tx input High requency clock "DIV2":data rate is 1/2 of the tx input high requency clock "DIV1":data rate is the same of the tx input high requency clock "MUL2":data rate is 2 times of the tx input high requency clock
PMA_REG_TX_RATE_EN	<string >	"FALSE", "TRUE"	"FALSE"	When reg_tx_rate_en is active high, ana_tx_rate_o<1:0> selects reg_tx_rate; otherwise, selects pma_tx_rate_i
PMA_REG_RX_TX2RX_ PLPBK_RST_N	<string >	"FALSE", "TRUE"	"TRUE"	CLK BLOCK reset over write
PMA_REG_RX_TX2RX_ PLPBK_RST_N_EN	<string >	"FALSE", "TRUE"	"FALSE"	CLK BLOCK reset over write ctrl
PMA_REG_RX_TX2RX_ PLPBK_EN	<string >	"FALSE", "TRUE"	"FALSE"	TX to RX parallel loopback enable
PMA_REG_RX_DATA_ POLARITY	<string >	"NORMAL"; "REVERSE"	"NORMAL"	Polarity reversal control signal, "REVERSE": 20bit data inversion output, "NORMAL": no operation
PMA_REG_RX_ERR_INSERT	<string >	"FALSE", "TRUE"	"FALSE"	Error injection control signal, when reg_rx_err_insert is configured as "TRUE", the output is the negation of the input; otherwise, the output data is the input input data
PMA_REG_UDP_CHK_EN	<string >	"FALSE", "TRUE"	"FALSE"	Data capture enable, data_i data will be captured to udp_o after configured as "TRUE"
PMA_REG_PRBS_SEL	<string >	"PRBS7"; "PRBS15"; "PRBS23"; "PRBS31"	"PRBS7"	Prbs mode selection, from reg_tx_prbs_sel "PRBS7": PRBS7; "PRBS15": PRBS15; "PRBS23": PRBS23; "PRBS31": PRBS31

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_PRBS_CHK_EN	<string >	"FALSE", "TRUE"	"FALSE"	Prbs chk enable
PMA_REG_LPLL_NFC_ STIC_DIS_N	<integer>	0, 1	0	When 1, lpll_nofbclk_sticky_o does not affect pll_locked_o; When 0, lpll_nofbclk_sticky_o is 1 pll_locked_o is 0.
PMA_REG_BIST_CHK_ PAT_SEL	<string >	"PRBS"; "CONSTANT"	"PRBS"	Select the mode of inspection, "PRBS": is prbs check, "CONSTANT": is udp check (udp check no use)
PMA_REG_LOAD_ERR_CNT	<string >	"FALSE", "TRUE"	"FALSE"	When "TRUE" err_counter_o remains unchanged, when "FALSE" err_counter_o is an internal error error statistics
PMA_REG_CHK_ COUNTER_EN	<string >	"FALSE", "TRUE"	"TRUE"	Error count enable
PMA_REG_CDR_PROP_ GAN_SEL	<integer>	0 to 7	3	Normal Mode Proportional Gain0: 1/2^9 1: 1/2^8 2: 1/2^7 3: 1/2^6 4: 1/2^5 5: 1/2^4 6: 1/2^3 7: 1/2^2
PMA_REG_CDR_TUBO_ PROP_GAIN_SEL	<integer>	0 to 7	6	turbo mode proportional gain 0: 1/2^9 1: 1/2^8 2: 1/2^7 3: 1/2^6 4: 1/2^5 5: 1/2^4 6: 1/2^3 7: 1/2^2
PMA_REG_CDR_INT_ GAIN_SEL	<integer>	0 to 7	2	Normal Mode Integral Gain Control:0: 1/2^16 1: 1/2^15 2: 1/2^14 3: 1/2^13 4: 1/2^12 5: 1/2^11 6: 1/2^10 7: 1/2^9
PMA_REG_CDR_TUBO_ INT_GAIN_SEL	<integer>	0 to 7	6	Turbo mode integral gain control: 0: 1/2^16 1: 1/2^15 2: 1/2^14 3: 1/2^13 4: 1/2^12 5: 1/2^11 6: 1/2^10 7: 1/2^9

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_CDR_INT_ SAT_MAX_4_0	<integer>	0 to 31	0	Integral upper limit, maximum -512, signed number
PMA_REG_CDR_INT_ SAT_MAX_9_5	<integer>	0 to 31	28	Integral upper limit, maximum -512, signed number
PMA_REG_CDR_INT_ SAT_MIN_2_0	<integer>	0 to 7	0	Integral lower limit, maximum 511, signed number
PMA_REG_CDR_INT_ SAT_MIN_9_3	<integer>	0 to 127	16	Integral lower limit, maximum 511, signed number
PMA_ANA_RX_REG_O_61_55	<integer>	0 to 127	21	61(bit6): From register reg_cdr_rx_pi_force_sel cdr rx PI control force value selection. 0: rx pi ctrl using the accumulator output. 1: rx pi ctrl using the force data(reg_cdr_rx_pi_force_d_i) only "cdr_rx_pi_ctrl_acc_o" is controlled by this signal 60(bit5): cdr lock counter reset overwrite, high valid. 0: disable. 1: reset the lock counter using reg_cdr_lock_rst 59(bit4): cdr lock counter reset, high valid. 0: disable. 1: reset the lock counter. 58(bit3): cdr proportional path reset overwrite, high valid. 0: disable. 1: reset the proportional path using reg_cdr_prop_rst 57(bit2): cdr proportional path reset, high valid 0: disable. 1: reset the proportional path. 56(bit1): cdr integral path reset overwrite, high valid 0: disable. 1: reset the integral path using reg_cdr_int_rst 55(bit0): cdr integral path reset, high valid 0: disable. 1: reset the integral path.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_ANA_RX_REG_O_69_62	<integer>	0 to 255	0	PI control code
PMA_ANA_RX_REG_O_77_70	<integer>	0 to 255	135	77(bit7): Integral path saturation detection enable: 0: disable; 1: enable. 76(bit6): cdr lock detection (cdr_ready) overwrite value. 75(bit5): cdr lock detection (cdr_ready) overwrite enable 0: disabled 1: cdr_lock is set by reg_cdr_lock_val. 74:73(bit4~bit3) 00: turbo_mode time is equal to the lock time. 01: turbo_mode time is (1/4 +1/2)of the lock time. 10: turbo_mode time is 1/2 of the lock time. 11: turbo_mode time is 1/4 of the lock time. "/" use floor 72:70(bit2~bit0) cdr lock detection pseudo timer control (for 312.5MHz clock) 000: 64cycles 001: 192cycles 010: 512cycles 011: 768cycles 100: 1024cycles 101: 1536cycles 110: 4096cycles 111: 8191cycles (others default: 111)

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_ANA_RX_REG_O_85_78	<integer>	0 to 255	1	89:81 (bit7~bit3, connected with the next parameter): cdr lock count value threshold, count value exceeding this threshold is considered locked 80:78(bit2~bit0) When cdr outputs the difference between two adjacent pi is less than this value, the count value counts.
PMA_ANA_RX_REG_O_93_86	<integer>	0 to 255	8	93 (bit7): cdr pi control reset overwrite. Active high. 0: disable. 1: reset the pi control using reg_cdr_pi_ctrl_rst 92 (bit6): cdr pictrl reset register 91 (bit5): When the cdr saturation is detected, reset the cdr: 0: disable; 1: enable. 90 (bit4): Enable cdr saturation status detection: 0: disable; 1: enable. 89:81 (bit3~bit0, connected with the previous parameter): cdr lock count value threshold, count value exceeding this threshold is considered locked

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_ANA_RX_REG_O_100_94	<integer>	0 to 127	64	100:99 (bit6~bit5): 00: 64 out ot 1024 cycles saturation detected is reckoned as saturation. 01: 128 out ot 1024 cycles saturation detected is reckoned as saturation. 10: 256 out ot 1024 cycles saturation detected is reckoned as saturation. 11: 511 out ot 1024 cycles saturation detected is reckoned as saturation. 98 (bit4): Disable the sigdet_status. 0: cdr reset when sigdet_status is low. 1: cdr donot response to sigdet_status transition. 97 (bit3): cdr saturation detection status sticky reset overwrite. Active high. 0: disable. 1: reset the sat_det status sticky using reg_cdr_sat_det_sticky_rst 96 (bit2): cdr saturation detection status sticky reset, Active high. 0: disable. 1: reset the sat_det status sticky. 95 (bit1): cdr saturation detection reset overwrite. Active high. 0: disable. 1: reset the sat_det using reg_cdr_sat_det_rst 94 (bit0): cdr saturation detection reset, Active high. 0: disable. 1: reset the sat_det module.
PMA_ANA_RX_REG_O_108_101	<integer>	0 to 255	0	108 (bit7): Enable signal of cdr status fifo 0: disabled 1: enabled 107:104 (bit6~bit3): reserved 103 (bit2): cdr turbo mode enable. 0: disable. 1: enable. 102 (bit1): cdr saturation detection status overwrite enabel. 0: disable 1: enable. cdr sat_det status using reg_cdr_sat_det_status_val 101 (bit0): register for cdr saturation detection status overwrite.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_ANA_RX_ REG_O_111_109	<integer>	0 to 7	0	Register reg_pma_test_sel Select which signal connected to fifo for test. 0: cdr pictrl value 1: cdr int data others: reserved
PMA_REG_OOB_ COMWAKE_GAP_MIN_4_0	<integer>	0 to 31	3	Low level minimum, comwake From reg_oob_comwake_gap_min
PMA_REG_OOB_ COMWAKE_GAP_MIN_5	<integer>	0, 1	0	Low level minimum, comwake From reg_oob_comwake_gap_min
PMA_REG_OOB_ COMWAKE_GAP_MAX	<integer>	0 to 63	11	Low level maximum, comwake reg_oob_comwake_gap_max
PMA_REG_OOB_ COMINIT_GAP_MIN	<integer>	0 to 255	15	Low level minimum, cominit From reg_oob_cominit_gap_min
PMA_REG_OOB_ COMINIT_GAP_MAX	<integer>	0 to 255	35	Low level maximum value, cominit From reg_oob_cominit_gap_max
PMA_REG_COMWAKE_ STATUS_CLEAR	<integer>	0, 1	0	Clear the comwake detected by the circuit, high level clear
PMA_REG_COMINIT_ STATUS_CLEAR	<integer>	0, 1	0	Clear the cominit monitored by the circuit, high level clear
PMA_REG_RX_SATA_ COMINIT_OW	<string >	"FALSE", "TRUE"	"FALSE"	When register reg_rx_sata_cominit_ow is "TRUE", pma_rx_sata_cominit_o comes directly from register reg_rx_sata_cominit for testing
PMA_REG_RX_SATA_ COMINIT	<string >	"FALSE", "TRUE"	"FALSE"	SATA signal status
PMA_REG_RX_SATA_ COMWAKE_OW	<string >	"FALSE", "TRUE"	"FALSE"	When register reg_rx_sata_comwake_ow is "TRUE", pma_rx_sata_comwake_o comes directly from register reg_rx_sata_comwake for testing
PMA_REG_RX_SATA_ COMWAKE	<string >	"FALSE", "TRUE"	"FALSE"	SATA signal status
PMA_REG_RX_DCC_DISABLE	<string >	"FALSE", "TRUE"	"FALSE"	Reserved
PMA_REG_RX_SLIP_SEL_EN	<string >	"FALSE", "TRUE"	"FALSE"	rx slip phase manually selection enable "FALSE": disable "TRUE": enable
PMA_REG_RX_SLIP_SEL_3_0	<integer>	0 to 15	0	rx slip phase manually selection(bit0~bit3), and PMA_REG_RX_SLIP_SEL_4 in combination. 0, N=0 1, N=1 2, N=2 ………. 19, N=19 others are not allowed

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_RX_SLIP_EN	<string >	"FALSE", "TRUE"	"FALSE"	Reserved
PMA_REG_RX_SIGDET_ STATUS_SEL	<integer>	0 to 7	5	rx signal detection status signals selection
PMA_REG_RX_SIGDET_ FSM_RST_N	<string >	"FALSE", "TRUE"	"TRUE"	signal det fsm reset signal
PMA_REG_RX_SIGDET_ STATUS	<string >	"FALSE", "TRUE"	"FALSE"	Monitor signal status value set by register
PMA_REG_RX_SIGDET_VTH	<string >	"27MV"; "9MV"; "18MV"; "36MV"; "45MV"; "54MV"; "63MV"; "72MV"	"36MV"	rx_sigdet threshold voltage selection "9MV": 9mV "18MV": 18mV "27MV": 27mV "36MV": 36mV "45MV": 45mV "54MV": 54mV "63MV": 63mV "72MV": 72mV
PMA_REG_RX_SIGDET_GRM	<integer>	0 to 3	0	rx_sigdet: filter bandwidth selection 0: high 1: media 2: media 3: low
PMA_REG_RX_SIGDET_ PULSE_EXT	<string >	"FALSE", "TRUE"	"FALSE"	rx_sigdet: filter bandwidth selection "FALSE": high "TRUE": low
PMA_REG_RX_SIGDET_ CH2_SEL	<integer>	0, 1	0	rx_sigdet_sel 0: Select glitch filtered waveform 1: Select the waveform before glitch filtering
PMA_REG_RX_SIGDET_ CH2_CHK_WINDOW	<integer>	0 to 31	3	clk window from deposit
PMA_REG_RX_SIGDET_ CHK_WINDOW_EN	<string >	"FALSE", "TRUE"	"TRUE"	clk window from deposit, overwrite ctrl signal
PMA_REG_RX_SIGDET_ NOSIG_COUNT_SETTIN G	<integer>	0 to 7	4	pma_rx_sigdet_status_o filter parameter N; 0: N=1 1: N=4 2: N=8 3: N=16 4: N=32 5: N=64 6: N=128 7: N=256

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_RX_SIGDET_ OOB_DET_COUNT_VAL_2_0	<integer>	0 to 7	0	rx_sigdet_status_2oob_det_o filter parameters
PMA_REG_RX_SIGDET_ OOB_DET_COUNT_VAL_4_3	<integer>	0 to 3	0	rx_sigdet_status_2oob_det_o filter parameters
PMA_REG_RX_SIGDET_ 4OOB_DET_SEL	<integer>	0 to 7	7	rx signal detection status signals selection
PMA_REG_RX_SIGDET_ IC_I	<integer>	0 to 15	10	Reserved
PMA_REG_RX_EQ1_ R_SET_TOP	<integer>	0 to 3	3	eq1 loading resistor 0: 150 ohm 1: 110 ohm 2: 95 ohm 3: 80 ohm
PMA_REG_RX_EQ1_ C_SET_FB	<integer>	0 to 15	0	Eq1 ac cap setting: 0: 180f 1: 564f … 7: 2820f others: Reserved
PMA_REG_RX_EQ1_OFF	<string >	"FALSE", "TRUE"	"FALSE"	eq off of ctle stage1
PMA_REG_RX_EQ2_ R_SET_TOP	<integer>	0 to 3	3	eq2 loading resistor 0: 150 ohm 1: 110 ohm 2: 95 ohm 3: 80 ohm
PMA_REG_RX_EQ2_ C_SET_FB	<integer>	0 to 15	0	Eq2 ac cap setting: 0: 180f 1: 564f … 7: 2820f others: Reserved
PMA_REG_RX_EQ2_OFF	<string >	"FALSE", "TRUE"	"FALSE"	eq off of ctle stage2
PMA_REG_RX_ICTRL_EQ	<integer>	0 to 3	2	Reserved
PMA_REG_EQ_DC_CALIB_EN	<string >	"FALSE", "TRUE"	"FALSE"	Reserved
PMA_CTLE_CTRL_REG_I	<integer>	0 to 15	0	Reserved
PMA_CTLE_REG_FORCE_ SEL_I	<string >	"FALSE", "TRUE"	"FALSE"	Reserved
PMA_CTLE_REG_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Reserved
PMA_CTLE_REG_INIT_ DAC_I_1_0	<integer>	0 to 3	0	Reserved
PMA_CTLE_REG_INIT_ DAC_I_3_2	<integer>	0 to 3	0	Reserved

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_CTLE_REG_POLARITY_I	<string >	"FALSE", "TRUE"	"FALSE"	Reserved
PMA_CTLE_REG_ SHIFTER_GAIN_I	<integer>	0 to 7	4	Reserved
PMA_CTLE_REG_ THRESHOLD_I_1_0	<integer>	0 to 3	0	Reserved
PMA_CTLE_REG_ THRESHOLD_I_9_2	<integer>	0 to 255	1	Reserved
PMA_CTLE_REG_ THRESHOLD_I_11_10	<integer>	0 to 3	0	Reserved
PMA_REG_RX_RES_TRIM_EN	<string >	"FALSE", "TRUE"	"FALSE"	When TRUE, pma_rx_term_res_trim_o comes from reg_rx_res_trim; when FALSE, from port
PMA_REG_ALG_RX_ TERM_POWER_DIVIDING_ SELECTION	<integer>	0 to 3	1	rx_term power dividing selection 0: 0.6v 1: 0.7v 2: 0.8v 3: 0.9v
PMA_REG_ALG_RX_ TERM_VCM_SELECTION	<integer>	0 to 7	3	rx_term vcm selection 0: 1.2V 1: 0V 2: floating 3: from internal buffer others: from power diving
PMA_REG_ALG_RX_ TERM_TEST_SELECTION_7_0	<integer>	0 to 255	0	rx_term test selection
PMA_REG_ALG_RX_ TERM_TEST_SELECTION_9_ 8	<integer>	0 to 3	0	rx_term test selection
PMA_REG_ALG_LOW_ SPEED_MODE_ENABLE	<string >	"FALSE", "TRUE"	"FALSE"	low speed mode enable "FALSE": disable "TRUE": enable
PMA_REG_ALG_RX_CLOCK_ POWER_DOWN_REGISTER	<string >	"FALSE", "TRUE"	"TRUE"	Rx clock power down register "FALSE": normal work "TRUE": power down
PMA_REG_ALG_RX_CLOCK_ POWER_DOWN_SELECTION	<string >	"FALSE", "TRUE"	"FALSE"	Rx clock power down selection "FALSE": pma_rx_pd "TRUE": pma_rx_reg<568>
PMA_REG_ALG_RX_DFE_ POWER_DOWN_REGISTER_0	<string >	"FALSE", "TRUE"	"TRUE"	Rx dfe power down register "FALSE": normal work "TRUE": power down
PMA_REG_ALG_RX_DFE_ POWER_DOWN_SELECTION_1	<string >	"FALSE", "TRUE"	"FALSE"	Rx dfe power down selection "FALSE": pma_rx_pd "TRUE": pma_rx_reg<571>

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_ALG_RX_CTLE_ POWER_DOWN_REGISTER_0	<string >	"FALSE", "TRUE"	"TRUE"	Rx Ctle power down register "FALSE": normal work "TRUE": power down
PMA_REG_ALG_RX_CTLE_ POWER_DOWN_SELECTION_1	<string >	"FALSE", "TRUE"	"FALSE"	Rx Ctle power down selection "FALSE": pma_rx_pd "TRUE": pma_rx_reg<572>
PMA_REG_EYE_DFETAP1_ PLORITY	<string >	"FALSE", "TRUE"	"FALSE"	EYE_DFETAP1_Plority: "TRUE": sel_pre_1; "FALSE": sel_pre_0;
PMA_REG_CDR_SEL	<string >	"FALSE", "TRUE"	"FALSE"	CDR_sel: "TRUE": 20bit data+cdrx sent to pcs "FALSE": 40bit data sent to pcs
PMA_REG_EYE_DET_EN	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": output the comparison result of eye data and data "FALSE": output eye data directly
PMA_REG_PI_BIAS_CURRENT	<integer>	0 to 3	0	pi fixed bias current setting
PMA_REG_ALG_DFE_ TEST_SEL_6_0	<integer>	0 to 127	0	dfe test selection
PMA_REG_ALG_DFE_ TEST_SEL_14_7	<integer>	0 to 255	0	dfe test selection
PMA_REG_ALG_DFE_ TEST_SEL_21_15	<integer>	0 to 127	0	dfe test selection
PMA_REG_ALG_EYE_ PATH_SEL	<integer>	0 to 1	0	reg_rx_pic_eye_path_sel 0: select cdr+eye_pic 1: select eye_pic
PMA_REG_ALG_CTLE_ TEST_SEL	<integer>	0 to 7	0	CTLE test selection
PMA_REG_RX_SLIP_ SEL_4	<string >	"FALSE", "TRUE"	"FALSE"	As bit4 and PMA_REG_RX_SLIP_SEL_3_0 constitute a 5bit parameter, "FALSE"=1'b0, "TRUE"=1'b1, rx slip phase manually selection: 00000, N=0 00001, N=1 00010, N=2 ……… 10011, N=19 others are not allowed
PMA_REG_ALG_RX_ T1_BUFF_EN	<string >	"FALSE", "TRUE"	"TRUE"	rx_t1_buff_en enable control
PMA_REG_ALG_RX_ CDRX_BUFF_EN	<string >	"FALSE", "TRUE"	"TRUE"	rx_cdrx_buff_en enable control
PMA_REG_ALG_RX_ VP_T1_SW_PLORITY	<string >	"FALSE", "TRUE"	"FALSE"	rx_vp_t1_sw_plority
PMA_REG_ALG_RX_ VP_PLORITY	<string >	"FALSE", "TRUE"	"TRUE"	rx_vp_plority

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_ALG_RX_ GAIN_CTRL_SUMMER	<string >	"FALSE", "TRUE"	"FALSE"	rx_gain_ctrl_summer
PMA_REG_ALG_RX_ DC_OFFSET_T1_EN	<string >	"FALSE", "TRUE"	"TRUE"	rx_dc_offset_t1_en
PMA_REG_ALG_RX_ DC_OFFSET_VP_EN	<string >	"FALSE", "TRUE"	"TRUE"	rx_dc_offset_vp_en
PMA_REG_ALG_RX_ DC_OFFSET_CDRX_EN	<string >	"FALSE", "TRUE"	"TRUE"	rx_dc_offset_cdrx_en
PMA_REG_ALG_RX_ DC_OFFSET_CDRY_EN	<string >	"FALSE", "TRUE"	"TRUE"	rx_dc_offset_cdry_en
PMA_REG_ALG_RX_ DC_OFFSET_EYE_EN	<string >	"FALSE", "TRUE"	"TRUE"	rx_dc_offset_eye_en
PMA_REG_ALG_SLICER_ DC_OFFSET_OVERWITE	<string >	"FALSE", "TRUE"	"FALSE"	slicer DC offset overwite "FALSE": select pma_rx_slicer_dccal_en_o "TRUE": Select pma_rx_core_reg<15>
PMA_REG_ALG_SLICER_ DC_OFFSET_REG	<string >	"FALSE", "TRUE"	"FALSE"	slicer DC offset reg "TRUE": enable "FALSE": disable
PMA_REG_RX_PGA_OFF	<string >	"FALSE", "TRUE"	"FALSE"	eq off of PGA
PMA_REG_ALG_CDR_ XWEIGHT_I	<integer>	0 to 4	4	X weight0: 0 1: 1 ... 4: 8 others: reserved
PMA_REG_ALG_CDR_ YWEIGHT_I	<integer>	0 to 4	4	Y weight0: 0 1: 1 ... 4: 8 others: reserved
PMA_REG_ALG_CTLE_ FLIPDIR_I	<string >	"FALSE", "TRUE"	"FALSE"	Error detection sign change "FALSE": The sign of the error is not changed "TRUE": The sign of the error is changed
PMA_REG_ALG_CTLE_ HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Totalizer works normally "TRUE": Keeps current state
PMA_REG_ALG_CTLE_ INITDAC_5_0	<integer>	0 to 63	0	initial DAC
PMA_REG_ALG_CTLE_ INITDAC_6	<integer>	0, 1	0	initial DAC
PMA_REG_ALG_CTLE_ OVERWREN_I	<string >	"FALSE", "TRUE"	"FALSE"	Manual mode enabled "FALSE": Adaptive mode "TRUE": Manual mode, output initdac
PMA_REG_ALG_CTLE_ SHIFT_I	<integer>	0 to 7	4	ctle integral gain 0: 1 1: 2 … 7: 128

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_ALG_CTLE_ TOPNUM_2_0	<integer>	0 to 7	4	dfe mode Ctle scan cut-off position, the default value is 20 (this parameter is set to 4, the next parameter is set to 2)
PMA_REG_ALG_CTLE_ TOPNUM_4_3	<integer>	0 to 3	2	dfe mode Ctle scan cut-off position, the default value is 20
PMA_REG_ALG_CTLEOFS_ FLIPDIR_I	<string >	"FALSE", "TRUE"	"TRUE"	Error detection sign change "FALSE": The sign of the error is not changed "TRUE": The sign of the error is changed
PMA_REG_ALG_CTLEOFS_ HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Totalizer works normally "TRUE": Keeps current state
PMA_REG_ALG_CTLEOFS_ INITDAC_3_0	<integer>	0 to 15	0	initial DAC
PMA_REG_ALG_CTLEOFS_ INITDAC_6_4	<integer>	0 to 7	4	initial DAC
PMA_REG_ALG_CTLEOFS_ OVERWREN	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Totalizer works normally "TRUE": Keeps current state
PMA_REG_ALG_CTLEOFS_ SHIFT_I	<integer>	0 to 7	4	ctle offset calibration integral gain 0: 1 1: 2 … 7: 128
PMA_REG_ALG_DFE_CTLE_ PWD	<string >	"FALSE", "TRUE"	"FALSE"	Disable dfe mode ctle "FALSE": enable ctle "TRUE": close ctle, output initdac
PMA_REG_ALG_H1_FLIPDIR_I	<string >	"FALSE", "TRUE"	"FALSE"	Error detection sign change "FALSE": The sign of the error is not changed "TRUE": The sign of the error is changed
PMA_REG_ALG_H1_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Totalizer works normally "TRUE": Keeps current state
PMA_REG_ALG_H1_ INITDAC_5_0	<integer>	0 to 63	0	initial DAC
PMA_REG_ALG_H1_ INITDAC_6	<integer>	0, 1	0	initial DAC
PMA_REG_ALG_H1_ OVERWREN_I	<string >	"FALSE", "TRUE"	"FALSE"	Manual mode enabled "FALSE": Adaptive mode "TRUE": Manual mode, output initdac
PMA_REG_ALG_H1_SHIFT_I	<integer>	0 to 7	4	h1 integral gain 0: 1 1: 2 … 7: 128

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_ALG_H2_FLIPDIR_I	<string >	"FALSE", "TRUE"	"FALSE"	Error detection sign change "FALSE": The sign of the error is not changed "TRUE": The sign of the error is changed
PMA_REG_ALG_H2_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Totalizer works normally "TRUE": Keeps current state
PMA_REG_ALG_H2_ INITDAC_0	<integer>	0, 1	0	initial DAC
PMA_REG_ALG_H2_ INITDAC_5_1	<integer>	0 to 31	0	initial DAC
PMA_REG_ALG_H2_ OVERWREN_I	<string >	"FALSE", "TRUE"	"FALSE"	Manual mode enabled "FALSE": Adaptive mode "TRUE": Manual mode, output initdac
PMA_REG_ALG_H2_ SHIFT_I_1_0	<integer>	0 to 3	0	h2 integral gain 0:1 1:2 ... 7:128
PMA_REG_ALG_H2_SHIFT_I_2	<integer>	0, 1	1	h2 integral gain 0:1 1:2 ... 7:128
PMA_REG_ALG_H3_FLIPDIR_I	<string >	"FALSE", "TRUE"	"FALSE"	Error detection sign change "FALSE": The sign of the error is not changed "TRUE": The sign of the error is changed
PMA_REG_ALG_H3_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Totalizer works normally "TRUE": Keeps current state
PMA_REG_ALG_H3_ INITDAC_4_0	<integer>	0 to 31	0	initial DAC
PMA_REG_ALG_H3_ INITDAC_5	<integer>	0, 1	1	initial DAC
PMA_REG_ALG_H3_ OVERWREN_I	<string >	"FALSE", "TRUE"	"FALSE"	Manual mode enabled "FALSE": Adaptive mode "TRUE": Manual mode, output initdac
PMA_REG_ALG_H3_SHIFT_I	<integer>	0 to 7	4	h3 integral gain 0:1 1:2 ... 7:128
PMA_REG_ALG_H4_FLIPDIR_I	<string >	"FALSE", "TRUE"	"FALSE"	Error detection sign change "FALSE": The sign of the error is not changed "TRUE": The sign of the error is changed
PMA_REG_ALG_H4_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Accumulator is working normally "TRUE": keep the current state

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_ALG_H4_INITDAC_0	<integer>	0, 1	0	initial DAC
PMA_REG_ALG_H4_INITDAC_4_1	<integer>	0 to 15	8	initial DAC
PMA_REG_ALG_H4_OVERWREN_I	<string >	"FALSE", "TRUE"	"FALSE"	Manual mode enabled "FALSE": Adaptive mode "TRUE": Manual mode, output initdac
PMA_REG_ALG_H4_SHIFT_I	<integer>	0 to 7	4	h4 integral gain 0: 1 1: 2 … 7: 128
PMA_REG_ALG_H5_FLIPDIR_I	<string >	"FALSE", "TRUE"	"FALSE"	Error detection sign change "FALSE": The sign of the error is not changed "TRUE": The sign of the error is changed
PMA_REG_ALG_H5_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Totalizer works normally "TRUE": Keeps current state
PMA_REG_ALG_H5_INITDAC	<integer>	0 to 31	16	initial DAC
PMA_REG_ALG_H5_OVERWREN_I	<string >	"FALSE", "TRUE"	"FALSE"	Manual mode enabled "FALSE": Adaptive mode "TRUE": Manual mode, output initdac
PMA_REG_ALG_H5_SHIFT_I	<integer>	0 to 7	4	h5 integral gain 0: 1 1: 2 … 7: 128
PMA_REG_ALG_H6_FLIPDIR_I	<string >	"FALSE", "TRUE"	"FALSE"	Error detection sign change "FALSE": The sign of the error is not changed "TRUE": The sign of the error is changed
PMA_REG_ALG_H6_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Totalizer works normally "TRUE": Keeps current state
PMA_REG_ALG_H6_INITDAC_2_0	<integer>	0 to 7	0	initial DAC
PMA_REG_ALG_H6_INITDAC_4_3	<integer>	0 to 3	2	initial DAC
PMA_REG_ALG_H6_OVERWREN_I	<string >	"FALSE", "TRUE"	"FALSE"	Manual mode enabled "FALSE": Adaptive mode "TRUE": Manual mode, output initdac
PMA_REG_ALG_H6_SHIFT_I	<integer>	0 to 7	4	h6 integral gain 0: 1 1: 2 … 7: 128

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_ALG_HCTLE_OFS_1_0	<integer>	0 to 3	0	High frequency ctle bias
PMA_REG_ALG_HCTLE_OFS_3_2	<integer>	0 to 3	2	High frequency ctle offset bias
PMA_REG_ALG_HCTLE_OVERWRDAC_5_0	<integer>	0 to 63	0	Manually write the hctle dac value
PMA_REG_ALG_HCTLE_OVERWRDAC_6	<integer>	0, 1	1	Manually write the hctle dac value
PMA_REG_ALG_HCTLE_OVERWREN	<string >	"FALSE", "TRUE"	"FALSE"	Manual mode enabled "FALSE": adaptive mode "TRUE": manual mode, output overwrdac
PMA_REG_ALG_LEQH_INITDAC_I	<integer>	0 to 15	0	initial dac
PMA_REG_ALG_LEQH_PWD_I	<string >	"FALSE", "TRUE"	"FALSE"	leah off signal "FALSE": enable the leah module "TRUE": disable the leah module
PMA_REG_ALG_LEQH_REG_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Totalizer works normally "TRUE": Keeps current state
PMA_REG_ALG_LEQH_REG_SHIFT_I	<integer>	0 to 7	4	leah integral gain 0: 1 1: 2 … 7: 128
PMA_REG_ALG_LEQL_INITDAC_I	<integer>	0 to 15	0	initial dac
PMA_REG_ALG_LEQL_PWD_I	<string >	"FALSE", "TRUE"	"FALSE"	leql close signal "FALSE": enable the leql module "TRUE": disable the leql module
PMA_REG_ALG_LEQL_REG_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Totalizer works normally "TRUE": Keeps current state
PMA_REG_ALG_LEQL_REG_PRESELECT_I	<integer>	0 to 15	7	Pre bit selection
PMA_REG_ALG_LEQL_REG_SHIFT_I	<integer>	0 to 7	4	leql integral gain0: 1 1: 2 … 7: 128
PMA_REG_ALG_NEXTBIT_I	<string >	"FALSE", "TRUE"	"FALSE"	define the next bit "FALSE": next bit is 0 "TRUE": next bit is 1
PMA_REG_ALG_SOFS_COUNTMAX_I_6_0	<integer>	0 to 127	127	ofs Calibrate the maximum value of the count, exceeding the maximum value
PMA_REG_ALG_SOFS_COUNTMAX_I_14_7	<integer>	0 to 255	255	done for calibration
PMA_REG_ALG_SOFS_COUNTMAX_I_19_15	<integer>	0 to 31	31	ofs Calibrate the maximum value of the count, exceeding the maximum value done for calibration

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_ALG_SOFS_DACWIN_I	<integer>	0 to 7	1	dac fluctuation threshold
PMA_REG_ALG_SOFS_FLIP_DIR_I	<string >	"FALSE", "TRUE"	"TRUE"	Error Sign Change Register "FALSE": do not change the sign of the error "TRUE": change the sign of the error
PMA_REG_ALG_SOFS_FORCE_I	<string >	"FALSE", "TRUE"	"FALSE"	Manual Write Control "FALSE": manual writing is not enabled "TRUE": manual writing is enabled, the dac output is forcedac
PMA_REG_ALG_SOFS_FORCEDAC_I_5_0	<integer>	0 to 63	0	Manual write enable is adopted dac
PMA_REG_ALG_SOFS_FORCEDAC_I_6	<integer>	0, 1	1	Manual write enable is adopted dac
PMA_REG_ALG_SOFS_FORCENUM_I	<integer>	0 to 15	0	Manually enter the serial number of the slicer, the serial number of the slicer and force configure the offset dac of a certain slicer together
PMA_REG_ALG_SOFS_INITDAC_2_0	<integer>	0 to 7	0	Initial DAC Register
PMA_REG_ALG_SOFS_INITDAC_6_3	<integer>	0 to 15	8	Initial DAC Register
PMA_REG_ALG_SOFS_SHIFT_I	<integer>	0 to 7	1	slicer offset calibration integral gain 0: 32 1: 64 … 7: 4096
PMA_REG_ALG_SOFS_SKIP_I	<string >	"FALSE", "TRUE"	"FALSE"	Skip ofs correction register "FALSE": do not skip sofs calibration "TRUE": skip sofs calibration, sofs_done outputs 1 directly
PMA_REG_ALG_SOFS_WINCOUNTMAX_I_7_0	<integer>	0 to 255	254	The dac counts the maximum value within the threshold, and the number of dac fluctuations exceeds the maximum value within the threshold, it is considered that the calibration of a certain slicer is completed.
PMA_REG_ALG_SOFS_WINCOUNTMAX_I_11_8	<integer>	0 to 15	15	The dac counts the maximum value within the threshold, and the number of dac fluctuations exceeds the maximum value within the threshold, it is considered that the calibration of a certain slicer is completed.
PMA_REG_ALG_ST_FLIPDIR_I	<string >	"FALSE", "TRUE"	"FALSE"	Error detection sign change "FALSE": The sign of the error is not changed "TRUE": The sign of the error is changed
PMA_REG_ALG_ST_FORCEN	<string >	"FALSE", "TRUE"	"FALSE"	Manual mode enabled "FALSE": adaptive mode "TRUE": manual mode, sliding tap position is tap_num, value is tap_dac
PMA_REG_ALG_ST_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Totalizer works normally "TRUE": Keeps current state
PMA_REG_ALG_ST_INITDAC_0	<integer>	0, 1	0	initial DAC

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_ALG_ST_ INITDAC_4_1	<integer>	0 to 15	8	initial DAC
PMA_REG_ALG_ST_RECALEN	<string >	"FALSE", "TRUE"	"FALSE"	Continuous adaptive control after traversal "FALSE": Do not enable continuous adaptive control after traversal "TRUE": enable continuous adaptation after traversal answer
PMA_REG_ALG_ST_SHIFT_I	<integer>	0 to 7	4	st integral gain 0:1 1:2 ... 7:128
PMA_REG_ALG_ST_ STARTCNT_7_0	<integer>	0 to 255	0	Start the waiting time of st, the value represents the waiting period
PMA_REG_ALG_ST_ STARTCNT_15_8	<integer>	0 to 255	128	Start the waiting time of st, the value represents the waiting period
PMA_REG_ALG_ST_ STARTCNT_19_16	<integer>	0 to 15	0	Start the waiting time of st, the value represents the waiting period
PMA_REG_ALG_ST_ TAPCNT_3_0	<integer>	0 to 15	0	One-time adaptive waiting time, the value represents the period
PMA_REG_ALG_ST_ TAPCNT_11_4	<integer>	0 to 255	0	One-time adaptive waiting time, the value represents the period
PMA_REG_ALG_ST_ TAPCNT_17_12	<integer>	0 to 63	2	One-time adaptive waiting time, the value represents the period
PMA_REG_ALG_ST_ TOPTAP_1_0	<integer>	0 to 3	3	st the maximum tap position, the maximum value is 15, corresponding to 4 tap positions 59, 60, 61, 62.
PMA_REG_ALG_ST_ TOPTAP_3_2	<integer>	0 to 3	3	Together with PMA_REG_ALG_ST_TOPTAP_1_0, it forms the maximum tap position of st, the maximum value is 15, corresponding to 4 tap positions 59, 60, 61, 62.
PMA_REG_ALG_SWCLK_DIV	<integer>	0 to 7	0	SWCLK divided clock
PMA_REG_ALG_TAPA_ DAC_3_0	<integer>	0 to 15	0	DAC digital code for tapa in manual mode
PMA_REG_ALG_TAPA_DAC_4	<integer>	0, 1	1	DAC digital code for tapa in manual mode
PMA_REG_ALG_TAPA_NUM	<integer>	0 to 63	7	The location of tapa in manual mode
PMA_REG_ALG_TAPB_DAC_0	<integer>	0, 1	0	DAC digital code of tapb in manual mode
PMA_REG_ALG_TAPB_ DAC_4_1	<integer>	0 to 15	8	DAC digital code of tapb in manual mode
PMA_REG_ALG_TAPB_ NUM_3_0	<integer>	0 to 15	8	The location of tapb in manual mode
PMA_REG_ALG_TAPB_ NUM_5_4	<integer>	0 to 3	0	The location of tapb in manual mode

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_ALG_TAPC_DAC	<integer>	0 to 31	16	DAC digital code of tapc in manual mode
PMA_REG_ALG_TAPC_NUM_0	<integer>	0, 1	1	The location of the tapc in manual mode
PMA_REG_ALG_TAPC_NUM_5_1	<integer>	0 to 31	4	The location of the tapc in manual mode
PMA_REG_ALG_TAPD_DAC_2_0	<integer>	0 to 7	0	DAC digital code of tapd in manual mode
PMA_REG_ALG_TAPD_DAC_4_3	<integer>	0 to 3	2	DAC digital code of tapd in manual mode
PMA_REG_ALG_TAPD_NUM	<integer>	0 to 63	10	The location of tapd in manual mode
PMA_REG_ALG_VP_FLIPDIR_I	<string >	"FALSE", "TRUE"	"FALSE"	Error detection sign change "FALSE": the error sign is not changed "TRUE": the error sign is changed
PMA_REG_ALG_VP_GRN_SHIFT_I	<integer>	0 to 7	5	grn integral gain 0: 1 1: 2 … 7: 128
PMA_REG_ALG_VP_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": the accumulator works normally "TRUE": keep the current state
PMA_REG_ALG_VP_IDEAL_2_0	<integer>	0 to 7	0	Adaptive target vp value
PMA_REG_ALG_VP_IDEAL_6_3	<integer>	0 to 15	10	Adaptive target vp value
PMA_REG_ALG_VP_INITDAC_I_3_0	<integer>	0 to 15	0	vp initial DAC
PMA_REG_ALG_VP_INITDAC_I_6_4	<integer>	0 to 7	0	vp initial DAC
PMA_REG_ALG_VP_OVERWREN	<string >	"FALSE", "TRUE"	"FALSE"	Manual mode enabled "FALSE": Adaptive mode "TRUE": Manual mode, output initdac
PMA_REG_ALG_VP_RED_SHIFT_I	<integer>	0 to 7	5	red integral gain 0: 1 1: 2 … 7: 128
PMA_REG_ALG_VPOFS_SEL_0	<integer>	0, 1	0	vp bias, the adaptive target is vp+-ofs interval
PMA_REG_ALG_VPOFS_SEL_2_1	<integer>	0 to 3	1	vp bias, the adaptive target is vp+-ofs interval
PMA_REG_ALG_H1_UPBOUND_5_0	<integer>	0 to 63	55	DFE_H1 upper limit value
PMA_REG_ALG_H1_UPBOUND_6	<integer>	0, 1	1	DFE_H1 upper limit value

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_ALG_CTLEOFS_PWDN	<string >	"FALSE", "TRUE"	"FALSE"	Turn off the ctle_ofs clock
PMA_REG_ALG_LEQH_OVEREN_I	<string >	"FALSE", "TRUE"	"FALSE"	leq mode hctle manual configuration enable "FALSE": adaptive "TRUE": manual configuration enabled
PMA_REG_ALG_LEQL_OVEREN_I	<string >	"FALSE", "TRUE"	"FALSE"	leq mode lctle manual configuration enable "FALSE": adaptive "TRUE": manual configuration enabled
PMA_REG_ALG_AGC_FLIPDIR_I	<string >	"FALSE", "TRUE"	"FALSE"	Error detection sign change "FALSE": The sign of the error is not changed "TRUE": change error sign
PMA_REG_ALG_AGC_HOLD_I	<string >	"FALSE", "TRUE"	"FALSE"	Accumulator hold enable signal "FALSE": Accumulator is working normally "TRUE": keep the current state
PMA_REG_ALG_AGC_INITDAC	<integer>	0 to 15	10	initial DAC
PMA_REG_ALG_AGC_LOWBOUND_1_0	<integer>	0 to 3	3	agc lower limit value, the agc adaptive result will not be less than the lower limit
PMA_REG_ALG_AGC_LOWBOUND_3_2	<integer>	0 to 3	0	agc lower limit value, the agc adaptive result will not be less than the lower limit
PMA_REG_ALG_AGC_OVERWREN_I	<string >	"FALSE", "TRUE"	"FALSE"	Manual mode enabled "FALSE": Adaptive mode "TRUE": Manual mode, output initdac
PMA_REG_ALG_AGC_PWD	<string >	"FALSE", "TRUE"	"FALSE"	agc adaptive off "FALSE": enable agc "TRUE": close agc, output initdac
PMA_REG_ALG_AGC_SHIFT_I	<integer>	0 to 7	4	agc integral gain 0: 1(2^0) 1: 2(2^1) …… 7: 128(2^7)
PMA_REG_ALG_AGC_UPBOUND_0	<integer>	0, 1	1	agc upper limit value, the agc adaptive result will not be greater than the upper limit
PMA_REG_ALG_AGC_UPBOUND_3_1	<integer>	0 to 7	7	agc upper limit value, the agc adaptive result will not be greater than the upper limit
PMA_REG_ALG_AGC_WAITSEL	<integer>	0 to 15	11	agc wait time selection, wait for a certain period of time to start self-adaptation 0: 2 1: 4 … 15: 65536
PMA_REG_PI_CTRL_SEL_RX	<string >	"FALSE", "TRUE"	"FALSE"	"TRUE": ana_rx_pic_eye_o<43:0> decoded from reg_pi_ctrl_rx<7:0> "FALSE": ana_rx_pic_eye_o<43:0> decoded from pma_rx_pic_eye_i<7:0>;
PMA_REG_PI_CTRL_RX_4_0	<integer>	0 to 31	0	rx pi_ctrl control code

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_PI_CTRL_RX_7_5	<integer>	0 to 7	0	rx pi_ctrl control code
PMA_CFG_RX_LANE_ POWERUP	<string >	"OFF"; "ON"	"ON"	Register setting pma_rx_powerup: "OFF": powerdown; "ON": powerup
PMA_CFG_RX_PMA_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	Register setting pma_rx_reset_n: "FALSE": reset; "TRUE": normal work
PMA_INT_PMA_RX_MASK_0	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to mask int triggered by pma prbs error
PMA_INT_PMA_RX_CLR_0	<string >	"FALSE", "TRUE"	"FALSE"	set as "TRUE" to clr int triggered by pma prbs error
PMA_CFG_CTLE_ADP_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	Register setting ctle_adp_rstn; "TRUE": normal work; "FALSE": reset
PMA_CFG_RX_CDR_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	rx_cdr reset; "TRUE": normal work; "FALSE": reset
PMA_CFG_RX_CLKPATH_ RSTN	<string >	"FALSE", "TRUE"	"TRUE"	rx_clkpath reset; "TRUE": normal work; "FALSE": reset
PMA_CFG_RX_DFE_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	rx_dfe reset; "TRUE": normal work; "FALSE": reset
PMA_CFG_RX_LEQ_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	rx_leq reset; "TRUE": normal work; "FALSE": reset
PMA_CFG_RX_SLIDING_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	rx_sliding reset; "TRUE": normal work; "FALSE": reset
PMA_CFG_RX_EYE_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	rx_eye reset; "TRUE": normal work; "FALSE": reset
PMA_CFG_RX_CTLE_ DCCAL_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	rx_ctle_dccal reset; "TRUE": normal work; "FALSE": reset
PMA_CFG_RX_SLICER_ DCCAL_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	rx_slicer_dccal reset; "TRUE": normal work; "FALSE": reset
PMA_CFG_RX_SLIP_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	rx_slip reset; "TRUE": normal work; "FALSE": reset
PMA_TX related parameters are as follows:				
PMA_REG_TX_BEACON_ TIMER_SEL	<integer>	0 to 3	0	Configure tx beacon timer selection 0: 3 clock cycles 1: 4 clock cycles 2: 8 clock cycles 3: 16 clock cycles

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_TX_BIT_CONV	<string >	"FALSE", "TRUE"	"FALSE"	Register configuration TX tx_bit_conv signal; "FALSE": data_o[19:0] is data_i[19:0], positive sequence "TRUE": data_o[19:0] is data_i[0:19],reverse order
PMA_REG_TX_RES_CAL	<integer>	0 to 63	50	register tx res ctrl code
PMA_REG_TX_UDP_DATA_20	<integer>	0 to 1	0	tx_udp_data
PMA_REG_TX_UDP_ DATA_26_21	<integer>	0 to 63	0	tx_udp_data
PMA_REG_TX_UDP_ DATA_34_27	<integer>	0 to 255	0	tx_udp_data
PMA_REG_TX_UDP_ DATA_39_25	<integer>	0 to 31	0	tx_udp_data
PMA_REG_TX_BUSWIDTH_EN	<integer>	0 to 1	0	TX buswidth enable
PMA_REG_TX_PD_POST	<string >	"OFF"; "ON"	"OFF"	register tx post cursor powerdown enable; "ON": tx post cursor not power down "OFF": tx post cursor powerdown
PMA_REG_TX_PD_POST_OW	<string >	"FALSE", "TRUE"	"FALSE"	When Reg_tx_PD_POST_OW is "true", select Reg_tx_pd_post; when reg_tx_pd_post_ow is "false" and reg_tx_pd_ow is "true"; when reg_tx_pd_post_ow is "fals E "and reg_tx_pd_ow is selected when" false " pma_pd_i||pll_pd_o||pma_tx_pd_i
PMA_REG_TX_BUSWIDTH	<string >	"8BIT", "10BIT", "16BIT", "20BIT", "32BIT", "40BIT"	"20BIT"	TX buswidth control; "8BIT": 8bit; "10BIT": 10bit; "16BIT": 16bit; "20BIT": 20bit; "32BIT": 32bit; "40BIT": 40bit.
PMA_REG_EI_PCLK_ DELAY_SEL	<integer>	0 to 3	0	Control tx_main_pre_z, tx_post_pre_z, tx_main_bdata, tx_post_bdata delay relative with normal parallel data. 0: 0 pclk cycle, the four signals are synchronized with parallel data. 1: -1 pclk cycle, the four signals are 1 cycle earlier than parallel data. 2: +1 pclk cycle, the four signals are 1 cycle later than parallel data. 3: reserved
PMA_REG_TX_AMP_DAC0	<integer>	0 to 63	25	tx full swing bias control, bit 5 reserved

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_TX_AMP_DAC1	<integer>	0 to 63	19	tx full swing bias control, bit 5 reserved
PMA_REG_TX_AMP_DAC2	<integer>	0 to 63	14	tx full swing bias control, bit 5 reserved
PMA_REG_TX_AMP_DAC3	<integer>	0 to 63	9	tx full swing bias control, bit 5 reserved
PMA_REG_TX_RXDET_ THRESHOLD	<string >	"84MV"; "28MV"; "56MV"; "112MV"	"84MV"	Tx rxdet threshold select; "28MV": vdda-28mV; "56MV": vdda-56mV; "84MV": vdda-84mV; "112MV": vdda-112mV;
PMA_REG_TX_BEACON_ OSC_CTRL	<string >	"FALSE", "TRUE"	"FALSE"	beacon osc frequnce ctrl output to tx analog block "TRUE": 18M "FALSE": 36M
PMA_REG_TX_PRBS_ GEN_WIDTH_SEL	<integer>	0 to 7	0	from reg_tx_prbs_gen_width_sel PRBS width select 0: 8 bit 1: 10 bit 2: 16 bit 3: 20 bit 4: 32bit 5: 40 bit
PMA_REG_TX_TX2RX_ SLPBACK_EN	<string >	"FALSE", "TRUE"	"FALSE"	Tx to rx serial loopback enable "FALSE": serial loopback disabled "TRUE": serial loopback enabled
PMA_REG_TX_PCLK_ EDGE_SEL	<string >	"FALSE", "TRUE"	"FALSE"	control whether rise edge or fall edge of txpclk is sent to digital interface. "FALSE": rise edge is selected. "TRUE": fall edge is selected.
PMA_REG_TX_PRBS_GEN_EN	<string >	"FALSE", "TRUE"	"FALSE"	Set Prbs enable
PMA_REG_TX_PRBS_SEL	<string >	"PRBS7"; "PRBS15"; "PRBS23"; "PRBS31"	"PRBS7"	Prbs mode selection, from reg_tx_prbs_sel "PRBS7": PRBS7 "PRBS15": PRBS15 "PRBS23": PRBS23 "PRBS31": PRBS31
PMA_REG_TX_UDP_ DATA_7_TO_0	<integer>	0 to 255	5	tx_udp_data
PMA_REG_TX_UDP_ DATA_15_TO_8	<integer>	0 to 255	235	tx_udp_data
PMA_REG_TX_UDP_ DATA_19_TO_16	<integer>	0 to 15	3	tx_udp_data
PMA_REG_TX_FIFO_ WP_CTRL	<integer>	0 to 7	4	tx_fifo write protection control signal (set the initial address for reading, the value is 0~5)
PMA_REG_TX_FIFO_EN	<string >	"FALSE", "TRUE"	"FALSE"	Register configuration tx_fifo_en enable

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_TX_DATA_ MUX_SEL	<integer>	0 to 3	0	Data selection control signal, parallel data mux select to select different data source 0: data from pcs 1: prbs bist data 2: udp data (reg_tx_udp_data) 3: rx loopback data
PMA_REG_TX_ERR_INSERT	<string >	"FALSE", "TRUE"	"FALSE"	Error injection control signal, when configured as "TRUE", the output is the inversion of the input (ie: error injection), otherwise the output data is the input data
PMA_REG_TX_SATA_EN	<string >	"FALSE", "TRUE"	"FALSE"	SATA mode enable register; "FALSE": SATA mode is disabled. "TRUE": SATA mode is enabled. This register is combined with pma_tx_ei_en_x to determine whether normal EI mode or OOB_EI mode valid only for SATA is active.
PMA_REG_RATE_CHANGE_ TXPCLK_ON_OW	<string >	"FALSE", "TRUE"	"FALSE"	rate change txpclk on overwrite signal "FALSE": rate change txpclk on is controlled by pma_rate_change_txpclk_on "TRUE": rate change txpclk on is controlled by reg_rate_change_txpclk_on
PMA_REG_RATE_CHANGE_ TXPCLK_ON	<string >	"FALSE", "TRUE"	"TRUE"	txpclk enable signal when rate changes "FALSE": txpclk is gated off during rate change "TRUE": txpclk is on
PMA_REG_TX_CFG_POST1	<integer>	0 to 31	0	Register Configuration tx_cfg_post1
PMA_REG_TX_CFG_POST2	<integer>	0 to 31	0	Register Configuration tx_cfg_post2
PMA_REG_TX_OOB_ DELAY_SEL	<integer>	0 to 15	0	controls the oob ei enable signal delay steps to compensate data latency. 0, 4, 8: 0 pclk cycle, tx_oob_ei_en is synchronized with parallel data. 1, 5, 9: -1 pclk cycle, tx_oob_ei_en is 1 cycle earlier than parallel data. 2, 6, 14: +1 pclk cycle, tx_oob_ei_en is 1 cycle later than parallel data. others: reserved
PMA_REG_TX_POLARITY	<string >	"NORMAL"; "RESERVE"	"NORMAL"	invert tx parallel data: "NORMAL": txpdata_out[19:0] = txpdata_in[19:0] "RESERVE": txpdata_out[19:0] = ~txpdata_in[19:0]
PMA_REG_TX_LS_MODE_EN	<string >	"FALSE", "TRUE"	"FALSE"	reg_tx_ls_mode_en low-speed 270M mode enable signal "FALSE": low-speed mode is disabled. "TRUE": low-speed mode is enabled.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_RX_JTAG_OE	<string >	"FALSE", "TRUE"	"TRUE"	Register configuration rx jtag module enable register "FALSE": disabled; "TRUE": enable
PMA_REG_RX_ACJTAG_ VHYSTSEL	<integer>	0 to 7	0	rx acjtag hysteresis voltage control register
PMA_REG_TX_RES_ CAL_EN	<string >	"FALSE", "TRUE"	"FALSE"	ana_tx_res_o comes from pll_res_trim_i when reg_tx_res_cal_en_i is "TRUE", otherwise from reg_tx_res_cal
PMA_REG_RX_TERM_ MODE_CTRL	<integer>	0 to 7	5	rx terminatin working mode control register 2: external DC, internal DC; 4: external DC, internal AC; 5: external AC, internal AC; 6: external AC, internal DC; others: reserved
PMA_REG_PLPBK_ TXPCLK_EN	<string >	"FALSE", "TRUE"	"FALSE"	txpclk enable register in parallel loopback mode "TRUE": rxpclk is used in parallel loopback mode to guarantee no FIFO overflow; "FALSE": txpclk is used (normal working mode).
PMA_REG_TX_PH_SEL_0	<integer>	0 to 1	1	TX PISO sample clock phase selection.
PMA_REG_TX_PH_SEL_6_1	<integer>	0 to 63	0	TX PISO sample clock phase selection.
PMA_REG_TX_CFG_PRE	<integer>	0 to 31	0	pre cursor tap coefficient.
PMA_REG_TX_CFG_MAIN	<integer>	0 to 63	0	main cursor tap coefficient.
PMA_REG_CFG_POST	<integer>	0 to 31	0	Configure reg_tx_cfg_pos register, when tx_deemp_s is, 00: ana_tx_cfg_post_o comes from reg_tx_cfg_post; 01: ana_tx_cfg_post_o comes from reg_tx_cfg_post1; 10: ana_tx_cfg_post_o comes from reg_tx_cfg_ post2; 11: ana_tx_cfg_post_o from reg_tx_cfg_post; tx_deemp_s is an internal signal, when reg_tx_deemp_ow_i is 1, tx_deemp_s comes from register reg_tx_deemp; when reg_tx_deemp_ow_i is 0, tx_deemp_s comes from port pma_tx_deemp_i

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_PD_MAIN	<string >	"FALSE", "TRUE"	"TRUE"	pd_pre/pd_main/pd_post for TX driver state selection
PMA_REG_PD_PRE	<string >	"FALSE", "TRUE"	"TRUE"	pd_pre/pd_main/pd_post for TX driver state selection
PMA_REG_TX_RXDET_TIMER_SEL	<integer>	0 to 255	87	register enable rx detect sample waiting time control
PMA_REG_TX_MOD_STAND_BY_EN	<string >	"FALSE", "TRUE"	"FALSE"	Enable stand_by mode; "FALSE": disable stand_by mode; "TRUE": enable stand_by mode.
PMA_REG_STATE_STAND_BY_SEL	<string >	"FALSE", "TRUE"	"FALSE"	select the output state when tx work in stand_by mode "FALSE": tx output logic 0 when stand_by mode; "TRUE": tx output logic 1 when stand_by mode.
PMA_REG_TX_SYNC_NEW	<string >	"FALSE", "TRUE"	"FALSE"	Register for TX lane synchronization control; "FALSE": no sync; "TRUE": sync
PMA_REG_TX_SYNC_NEW_OW	<string >	"FALSE", "TRUE"	"FALSE"	TX lane synchronization control overwrite signal; "FALSE": TX lane synchronization is controlled by PCS; "TRUE": TX lane synchronization is controlled by reg_tx_sync_new
PMA_REG_TX_CHANGE_ON_OW	<string >	"FALSE", "TRUE"	"FALSE"	Rate_change_txpclk_on overwrite signal "FALSE": txpclk enable signal is controlled by rate_change_txpclk_on; "TRUE": txpclk enable signal is controlled by reg_tx_change_on_en
PMA_REG_TX_CHANGE_ON_POLAR_CTRL	<string >	"FALSE", "TRUE"	"FALSE"	invert tx_rate_change_on signal; "FALSE": no invert "TRUE": invert
PMA_REG_TX_FREERUN_PD	<string >	"FALSE", "TRUE"	"TRUE"	tx_freerun pd signal; "FALSE": don't power down tx_freerun; "TRUE": power down tx_freerun
PMA_REG_TX_CHANGE_ON_SEL	<string >	"FALSE", "TRUE"	"FALSE"	tx_rate_change_on signal selection; "FALSE": select pma_tx_rate_change_on0_i; "TRUE": select pma_tx_rate_change_on1_i
PMA_REG_TX_CHANGE_ON_CTRL	<string >	"FALSE", "TRUE"	"FALSE"	Register for pma_txpclk_freerun enable signal when rate changes
PMA_REG_TX_FREERUN_RATE_0	<integer>	0 to 1	0	Register configuration tx_freerun_rate

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_TX_FREERUN_ RATE_1	<integer>	0 to 1	0	Register configuration tx_freerun_rate
PMA_REG_TX_FREERUN_ RATE_OW	<string >	"FALSE", "TRUE"	"FALSE"	When reg_tx_freerun_rate_ow is "TRUE", reg_tx_freerun_rate overrides pma_tx_rate
PMA_REG_TX_RST_SYNC_ CLK_SEL	<string >	"FALSE", "TRUE"	"TRUE"	Select the clock for synchronizing the reset signal of tx; "FALSE": select pll_refclk; "TRUE": select tx_clk_in.
PMA_REG_TX_PI_CTRL_SEL	<integer>	0 to 3	0	pi control selection for tx
PMA_REG_TX_PI_CTRL	<integer>	0 to 255	0	pi control code for tx
PMA_LANE_POWERUP	<string >	"FALSE", "TRUE"	"TRUE"	Configure pma lane powerup
PMA_POR_N	<string >	"FALSE", "TRUE"	"TRUE"	configure pma lane resetn
PMA_TX_LANE_POWERUP	<string >	"FALSE", "TRUE"	"TRUE"	Configure tx lane driver powerup
PMA_TX_PMA_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	Configure tx lane resetn
PMA_LPLL_POWERUP	<string >	"FALSE", "TRUE"	"TRUE"	configure pma lpll powerdown
PMA_LPLL_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	configure pma lpll resetn
PMA_LPLL_LOCKDET_RSTN	<string >	"FALSE", "TRUE"	"TRUE"	configure pma lpll lock delete resetn
PMA_REG_LPLL_ PFDDELAY_SEL	<integer>	0 to 3	1	PLL PFD delay sel
PMA_REG_LPLL_ PFDDELAY_EN	<string >	"FALSE", "TRUE"	"TRUE"	PLL PFD delay enable; "TRUE": enable; "FALSE": disable
PMA_REG_LPLL_VCTRL_SET	<integer>	0 to 3	0	Pll vctrl start voltage
PMA_LPLL_CHARGE_ PUMP_CTRL	<string >	"type"; "min"; "max"	"type"	pll charge pump current control "min": min current; "type": typ current; "max": max current;
PMA_LPLL_REFDIV	<string >	"DIV2"; "DIV3"; "DIV4_0"; "DIV5"; "DIV4_1"; "DIV6"; "DIV8_0"; "DIV10"; "DIV8_1"; "DIV12"; "DIV16"; "DIV20"; "DIV1"	"DIV1"	Pll reference clock divider M pll_refdiv<4:0> div ratio "DIV1": 1; "DIV2": 2; "DIV3": 3; "DIV4_0": 4; "DIV4_1": 4; "DIV5": 5; "DIV6": 6; "DIV8_0": 8; "DIV8_1": 8; "DIV10": 10; "DIV12": 12; "DIV16": 16; "DIV20": 20; others: not allowed.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_LPLL_FBDIV	<integer>	0 to 7, 12 to 39, 44 to 63	38	pll_fbdiv<5:0> N2 N1 div ratio frequency division ratio; 0: 2 4 8; 1: 3 4 12; 2: 4 4 16; 3: 5 4 20; 4: 4 4 16; 5: 6 4 24; 6: 8 4 32; 7: 10 4 40; 8~11: Not allowed; 12: 8 4 32; 13: 12 4 48; 14: 16 4 64; 15: 20 4 80; 16~31: 1 4 4; 32: 2 5 10; 33: 3 5 15; 34: 4 5 20; 35: 5 5 25; 36: 4 5 20; 37: 6 5 30; 38: 8 5 40; 39: 10 5 50; 40~43: Not allowed; 44: 8 5 40; 45: 12 5 60; 46: 16 5 80; 47: 20 5 100; 48~63: 1 5 5
PMA_LPLL_LPF_RES	<integer>	0 to 15	1	LPF resistor control Default: lpf_res<3:0>=0001
PMA_LPLL_REFLOSS_READY	<integer>	0 to 1	0	Selection configuration for Refclk_loss_b & ana_lpll_ready_o output and ana_lpll_ready_o output alone: 0: Select Refclk_loss_b & ana_lpll_ready_o output; 1: Select ana_lpll_ready_o to output separately;
PMA_LPLL_LOCKED_PFDDELAY	<integer>	0 to 1	0	cplllocked signal selection: 0: select the txrx_interface ana_lpll_ready_o controls cplllocked; 1: The cplllocked signal is connected to low potential;
PMA_LPLL_MCLK_SEL	<integer>	0 to 1	0	sel_osc_mclk 0: osc_mclk output 36MHZ; 1: sel_osc_mclk output 18Mhz;
PMA_LPLL_TEST_SEL	<integer>	0 to 31	0	atb test

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_LPLL_TEST_SIG_ HALF_EN	<string >	"FALSE", "TRUE"	"TRUE"	test signal voltage half enable; "FALSE": test signal voltage out is 1:1; "TRUE": test signal voltage out is 1:3
PMA_LPLL_TEST_V_EN	<string >	"FALSE", "TRUE"	"FALSE"	test signal voltage enable; "FALSE": test signal of voltage; "TRUE": test signal of voltage is enabled
PMA_LPLL_MCLK_EN	<string >	"FALSE", "TRUE"	"TRUE"	en_osc_mclk "FALSE": disable "TRUE": enable（default）
PMA_LPLL_MCLK_DET_CTL	<integer>	0 to 31	16	refclk_det_ct<4:0> refers to the osc_clk count cycle: 31:2^4 30:2^5 28:2^6 24:2^7 16:2^8(default) 0:2^9;
PMA_LPLL_LOCKDET_REFCT	<integer>	0 to 7	3	lpll Reference clock domain counter overflow value Counter value for reference clock time out value of lock detect 0: 2^9 1: 2^10 2: 2^11 3: 2^12 4: 2^13 5: 2^14 6: 2^15 7: 2^16-1 (default)
PMA_LPLL_LOCKDET_FBCT	<integer>	0 to 7	3	Feedback clock domain counter overflow value counter value for feedback clock time out value of lock detect 0: 2^9 1: 2^10 2: 2^11 3: 2^12 4: 2^13 5: 2^14 6: 2^15 7: 2^16-1 (default)
PMA_LPLL_LOCKDET_LOCKCT	<integer>	0 to 7	4	Difference counter overflow value Register value of lock detection threshold for count difference between ref_clk & fb_clk counters. Default=100 0: 2 1: 4 2: 8 3: 16 4: 32 (default) 5: 64 6: 128 7: 256

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_LPLL_LOCKDET_ITER	<integer>	0 to 7	1	lock cycle times Register value of number of iterations of consecutive successful one-time lock detection before set pll_lock. 0: 1 1: 2 2: 4 3: 8(default) 4: 16 5: 32 6: 64 7:127
PMA_LPLL_UNLOCKDET_ITER	<integer>	0 to 3	2	Number of non-locking cycles Register value of number of iterations of lock detection attempts before set pll_unlock. 0: 64 1: 128 2: 256 (default) 3: 1023
PMA_LPLL_LOCKDET_EN_OW	<integer>	0; 1	0	Lockdet en selection control When 1, the internal pll_lockdet_en uses reg_lpll_lockdet_en_i; When 0, the internal pll_lockdet_en uses pll_lockdet_en_i
PMA_LPLL_LOCKDET_EN	<integer>	0; 1	0	Lockdet en for register input 0:disable 1: Enable
PMA_LPLL_LOCKDET_MODE	<integer>	0; 1	0	Lockdet mode 0:counter mode; 1:timer mode
PMA_LPLL_LOCKDET_OW	<integer>	0; 1	0	Locked signal output selection 1: pll_locked_o uses reg_lpll_locked_i; 0: pll_locked_o uses pll_locked_r (module internal status output)
PMA_LPLL_LOCKDETED	<integer>	0; 1	0	Locked signal for register input 1: locked; 0: unlocked
PMA_LPLL_UNLOCKDET_OW	<integer>	0; 1	0	unlocked signal output selection 1: unpll_locked_o use reg_lpll_unlocked; 0: unpll_locked_o use pll_locked_r (module internal status output)

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_LPLL_UNLOCKDETED	<integer>	0; 1	0	unlocked signal for register input 1: unlocked; 0: locked
PMA_LPLL_LOCKED_ STICKY_CLEAR	<integer>	0; 1	0	LPLL continuous lock signal clear signal 1: clear pll_locked_sticky_o; 0: not clear
PMA_LPLL_UNLOCKED_ STICKY_CLEAR	<integer>	0; 1	0	LPLL continuous loss of lock signal clear signal 1: clear pll_unlocked_sticky_o; 0: not clear
PMA_LPLL_LOCKDET_REPEAT	<integer>	0; 1	0	Retest after lock 1: enable; 0: disable
PMA_LPLL_NOFBCLK_ STICKY_CLEAR	<integer>	0; 1	0	lpll_nofbclk_sticky_o clear signal 1:clear; 0:unclear
PMA_LPLL_NOREFCLK_ STICKY_CLEAR	<integer>	0; 1	0	lpll_norefclk_sticky_o clear signal 1:clear; 0:unclear
PMA_LPLL_READY_OR_LOCK	<integer>	0; 1	0	Lock and ready options: 1: Select lpll_lock_detect output pll_locked_o; 0: pll_reaby output by lpll_fsm
PMA_LPLL_READY	<integer>	0; 1	0	lpll_ready signal for register input 1: ready; 0: unready
PMA_LPLL_READY_OW	<integer>	0; 1	0	lpll_ready signal output selection 1: lpll_ready use reg_lpll_ready; 0: lpll_ready Use lpll_ready (module internal status output)
PMA_REG_TXCLK_SEL	<string >	"LPLL"; "HPLL"	"HPLL"	Tx clock selction: Lpll or Hpll
PMA_REG_RXCLK_SEL	<string >	"LPLL"; "HPLL"	"HPLL"	Rx clock selction: Lpll or Hpll
PMA_REG_TEST_BUF	<integer>	0 to 7	0	test for channel buffer
PMA_REG_CHL_TEST	<integer>	0 to 3	0	a test for channel
PMA_REG_CHL_BIAS_ POWER_SEL	<string >	"FALSE"; "TRUE"	"FALSE"	channel bias power signal selection "FALSE": select ana_lane_reg_o<67>; "TRUE": select ana_pd_o
PMA_REG_CHL_BIAS_POWER	<string >	"FALSE"; "TRUE"	"FALSE"	channel bias power signal "FALSE": power on; "TRUE": power down
PMA_REG_IBIAS_DYNAMIC_ PD_7_0	<integer>	[255, 0]	0	Corresponding register reg_ibup_dyn_pd<8:1>, ibias dynamic current power down control; 1: power down; 0: work
PMA_REG_IBIAS_DYNAMIC_ PD_15_8	<integer>	[255, 0]	0	Corresponding register reg_ibup_dyn_pd<16:9>, ibias dynamic current power down control; 1: power down; 0: work
PMA_REG_IBIAS_DYNAMIC_ PD_18_16	<integer>	[7, 0]	0	Corresponding to reg_ibup_dyn_pd<19:17> registered device, ibias dynamic current power down control; 1: power down; 0: work

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_IBIAS_STA_CUR_PD_2_0	<integer>	[7, 0]	0	Corresponding to reg_ibup_sta_pd<3:1> register, ibias static current power down control; 1: power down; 0: work
PMA_REG_IBIAS_STA_CUR_PD_8_3	<integer>	[63, 0]	0	Corresponding to reg_ibup_sta_pd<9:4> register, ibias static current power down control; 1: power down; 0: work
PMA_REG_BANDGAP_VOL_SEL	<string>	"BANDGAP"; "VDDA"	"BANDGAP"	bandgap voltage sel; "BANDGAP": bandgap; "VDDA": vdda
PMA_REG_BANDGAP_TEST	<integer>	[15, 0]	0	bandgap test
PMA_REG_RX_PIBUF_IC	<integer>	[3, 0]	3	reg_rx_dcc_ic_rx[3]+reg_rx_pibuf_ic rx pi buf bandwidth control logic 3'b000: 1GHz 3'b001: 2GHz 3'b010: 4GHz 3'b011: 5GHz 3'b100: no use 3'b101: no use 3'b110: no use 3'b111: 10GHz
PMA_REG_RX_DCC_IC_ RX	<integer>	[3, 0]	1	[4]: rx 2nd PI enable [3]+reg_rx_pibuf_ic rx pi buf bandwidth control logic 3'b000: 1GHz 3'b001: 2GHz 3'b010: 4GHz 3'b011: 5GHz 3'b100: no use 3'b101: no use 3'b110: no use 3'b111: 10GHz
PMA_REG_TX_PI_CUR_B UF	<string>	"1GHz", "2GHz", "4GHz", "5GHz", "10GHz"	"10GHz"	tx pi buf bandwidth control logic
PMA_REG_CLK_SEL_ STROBE_TXPCLK	<integer>	[3, 0]	2	TX PISO sample clock phase selection.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_RX_DEF_SEL0	<string>	"25uA", "18.75uA", "31.25uA", "37.5u A"	"18.75uA"	Rx dfe current selection
PMA_REG_RX_DEF_SEL1	<string>	"25uA", "18.75uA", "31.25uA", "37.5uA"	"18.75uA"	Rx dfe current selection
PMA_REG_RX_EQ0_SEL	<string>	"50uA", "43.75uA", "56.25uA", "62.5uA"	"56.25uA"	Rx eq0 current selection
PMA_REG_RX_EQ1_SEL	<string>	"50uA", "43.75uA", "56.25uA", "62.5uA"	"56.25uA"	Rx eq1 current selection
PMA_REG_RX_PGA_SEL	<string>	"50uA", "43.75uA", "56.25uA", "62.5uA"	"56.25uA"	Rx PGA circuit current selection
PMA_REG_TX_DIV_SEL	<string>	"25uA", "18.75uA", "31.25uA", "37.5uA"	"25uA"	Tx driver current selection
PMA_REG_LPLL_AMP_SEL	<string>	"25uA", "18.75uA", "31.25uA", "37.5uA"	"25uA"	Lpll amp current selection
PMA_REG_LPLL_VCO_SEL	<string>	"25uA", "18.75uA", "31.25uA", "37.5uA"	"25uA"	Lpll vco current selection
PMA_REG_LPLL_CHARGE_PUMP_SEL	<string>	"25uA", "18.75uA", "31.25uA", "37.5uA"	"25uA"	Lpll charge pump current selection
PMA_REG_IBIAS_STATIC_SEL_7_0	<integer>	[255, 0]	255	reg_ibup_dyn_a<8:1> ibias static current selection: 0: 1X/3X 1: 2X/4X (1X current=25uA)
PMA_REG_IBIAS_STATIC_SEL_15_8	<integer>	[255, 0]	247	reg_ibup_dyn_a<16:9> ibias static current selection: 0: 1X/3X 1: 2X/4X (1X current=25uA)
PMA_REG_IBIAS_STATIC_SEL_18_16	<integer>	[7, 0]	7	reg_ibup_dyn_a<19:17> ibias static current selection: 0: 1X/3X 1: 2X/4X (1X current=25uA)
PMA_REG_IBIAS_STATIC_SEL_21_19	<integer>	[7, 0]	0	reg_ibup_dyn_b<3:1> ibias static current selection: 0: 1X/2X 1: 3X/4X (1X current=25uA)

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_IBIAS_STATIC_SEL_29_22	<integer>	[255, 0]	0	reg_ibup_dyn_b<11:4> ibias static current selection: 0: 1X/2X 1: 3X/4X (1X current=25uA)
PMA_REG_IBIAS_STATIC_SEL_37_30	<integer>	[255, 0]	1	reg_ibup_dyn_b<19:12> ibias static current selection: 0: 1X/2X 1: 3X/4X (1X current=25uA)
PMA_REG_IBIAS_STA_CUR_SEL_4_0	<integer>	[31, 0]	31	reg_ibup_sta_a<5:1> ibias static current selection: 0: 1X/3X 1: 2X/4X (1X current=12.5uA)
PMA_REG_IBIAS_STA_CUR_SEL_8_5	<integer>	[15, 0]	15	reg_ibup_sta_a<9:6> ibias static current selection: 0: 1X/3X 1: 2X/4X (1X current=12.5uA)
PMA_REG_IBIAS_STA_CUR_SEL_12_9	<integer>	[15, 0]	0	reg_ibup_sta_b<4:1> ibias static current selection: 0: 1X/2X 1: 3X/4X (1X current=12.5uA)
PMA_REG_IBIAS_STA_CUR_SEL_17_13	<integer>	[31, 0]	0	reg_ibup_sta_b<9:5> ibias static current selection: 0: 1X/2X 1: 3X/4X (1X current=12.5uA)

Note: 1. The parameter value FALSE, TRUE corresponds to the signal value 0, 1;

2. The parameters with values of "ON" and "OFF" are all power control parameters. Depending on whether the parameter is powderdown or powerup, the corresponding SC conversion is different. But the parameter value "ON" corresponds to power and "OFF" corresponds to no power.

5. Instantiation template

```
GTP_HSSTHP_LANE #(
.PCS_DYN_DLY_SEL_RX("FALSE"),
.PCS_PMA_RCLK_POLINV("PMA_RCLK"),
.PCS_PCS_RCLK_SEL("RCLK"),
.PCS_GEAR_RCLK_SEL("RCLK"),
.PCS_RCLK2FABRIC_SEL("HARD_1"),
.PCS_SCAN_INTERVAL_RX("4_CLOCKS"),
.PCS_BRIDGE_RCLK_SEL("RCLK"),
.PCS_RCLK_POLINV("RCLK"),
.PCS_TO_FABRIC_CLK_SEL("PMA_RCLK"),
.PCS_CLK2ALIGNER_SEL("TO_FABRIC_CLK"),
```

```
.PCS_TO_FABRIC_CLK_DIV_EN("FALSE"),
.PCS_AUTO_NEAR_LOOP_EN("FALSE"),
.PCS_PCS_RCLK_EN("FALSE"),
.PCS_BRIDGE_PCS_RCLK_EN_SEL("HARD_1"),
.PCS_BRIDGE_RCLK_EN_SEL("HARD_0"),
.PCS_GEAR_RCLK_EN_SEL("HARD_0"),
.PCS_NEGEDGE_EN_RX("FALSE"),
.PCS_PCS_RX_RSTN("FALSE"),
.PCS_BRIDGE_PCS_RSTN("FALSE"),
.PCS_TO_FABRIC_RST_EN("FALSE"),
.PCS_BYPASS_GEAR_RRSTN("FALSE"),
.PCS_BYPASS_BRIDGE_RRSTN("FALSE"),
.PCS_ALIGNER_EN_RX("FALSE"),
.PCS_RX_SLAVE("MASTER"),
.PCS_RX_CA(0),
.PCS_SUM_THRESHOLD_RX(0),
.PCS_AVG_CYCLES_RX(0),
.PCS_REG_PMA_RX2TX_PLOOP_EN("FALSE"),
.PCS_REG_PMA_RX2TX_PLOOP_FIFOEN("FALSE"),
.PCS_STEP_SIZE_RX(0),
.PCS_REV_CNT_LIMIT_RX(0),
.PCS_FILTER_CNT_SIZE_RX(0),
.PCS_DLY_ADJUST_SIZE_RX_3_0(0),
.PCS_DLY_REC_SIZE_RX(0),
.PCS_ALIGN_THRD_RX(0),
.PCS_DLY_ADJUST_SIZE_RX_4(0),
.PCS_CFG_DEC_TYPE_EN("FALSE"),
.PCS_RXBRIDGE_GEAR_SEL("FALSE"),
.PCS_GE_AUTO_EN("FALSE"),
.PCS_RXBRG_FULL_CHK_EN("FALSE"),
.PCS_RXBRG_EMPTY_CHK_EN("FALSE"),
.PCS_IFG_EN("FALSE"),
.PCS_FLP_FULL_CHK_EN("FALSE"),
.PCS_FLP_EMPTY_CHK_EN("FALSE"),
.PCS_RX_POLARITY_INV("DELAY"),
.PCS_FARLP_PWR_REDUCTION("FALSE"),
.PCS_RXPRBS_PWR_REDUCTION("NORMAL"),
.PCS_WDALIGN_PWR_REDUCTION("NORMAL"),
.PCS_RXDEC_PWR_REDUCTION("NORMAL"),
.PCS_RXBRG_PWR_REDUCTION("NORMAL"),
.PCS_RXTEST_PWR_REDUCTION("NORMAL"),
.PCS_WA_SOS_DET_TOL(0),
.PCS_WA_SE_DET_TOL(0),
.PCS_RX_SAMPLE_UNION("FALSE"),
.PCS_NEAR_LOOP("FALSE"),
.PCS_BYPASS_WORD_ALIGN("FALSE"),
.PCS_BYPASS_DENC("FALSE"),
.PCS_RX_ERRCNT_CLR("FALSE"),
```

```
.PCS_RX_CODE_MODE("DUAL_8B10B"),
.PCS_RX_BYPASS_GEAR("FALSE"),
.PCS_ERRDETECT_SILENCE("FALSE"),
.PCS_RX_DATA_MODE("8BIT"),
.PCS_CA_DYN_CLY_EN_RX("FALSE"),
.PCS_CFG_APATTERN_STATUS_DELAY("DELAY_ONE_CYCLE"),
.PCS_RX_PRBS_MODE("DISABLE"),
.PCS_ALIGN_MODE("1GB"),
.PCS_COMMA_DET_MODE("PATTERN_DETECT"),
.PCS_RAPID_VMIN_1(0),
.PCS_RAPID_VMIN_2(0),
.PCS_RXBU_WIDER_EN("40/20BIT"),
.PCS_RAPID_IMAX(0),
.PCS_RX_SPLIT("SPLIT_22BIT_11BIT"),
.PCS_RXBRG_END_PACKET_9_8(0),
.PCS_RXBRG_END_PACKET_7_0(0),
.PCS_CTC_MAX_DEL(0),
.PCS_COMMA_REG0_9_8(0),
.PCS_COMMA_REG1_9_8(0),
.PCS_COMMA_MASK_9_8(0),
.PCS_COMMA_REG0_7_0(0),
.PCS_COMMA_REG1_7_0(0),
.PCS_COMMA_MASK_7_0(0),
.PCS_FLP_WRADDR_START(0),
.PCS_FLP_RDADDR_START(0),
.PCS_CFG_RX_BRIDGE_CLK_POLINV("FALSE"),
.PCS_CTC_MODE_RD_SEL("NOMINAL_EMPTY"),
.PCS_CTC_AFULL(0),
.PCS_FAST_LOCK_GEAR_EN("FALSE"),
.PCS_CTC_MODE_WR_SEL("NOMINAL_EMPTY"),
.PCS_CTC_AEMPTY(0),
.PCS_CTC_MODE("ONE_BYTE"),
.PCS_RXBRIDGE_MODE("BYPASS"),
.PCS_CTC_ADD_MAX(0),
.PCS_CFG_PHDET_EN_RX("FALSE"),
.PCS_WA_SDS_DET_TOL(0),
.PCS_CEB_MODE("10GB"),
.PCS_APATTERN_MODE("ONE_BYTE"),
.PCS_A_REG0_8("FALSE"),
.PCS_RXBRG_WADDR_START(0),
.PCS_A_REG1_8("FALSE"),
.PCS_RXBRG_RADDR_START(0),
.PCS_A_REG0_7_0(0),
.PCS_A_REG1_7_0(0),
.PCS_CEB_RAPIDLS_MMAX(0),
.PCS_CEB_DETECT_TIME(0),
.PCS_WL_FIFO_RD(0),
.PCS_SKIP_REG0_9_8(0),
```

```
.PCS_SKIP_REG0_7_0(0),
.PCS_CFG_CONTI_SKP_SET(0),
.PCS_CFG_RX_BASE_ADV_MODE("BASE_MODE"),
.PCS_SKIP_REG1_9_8(0),
.PCS_SKIP_REG2_9_8(0),
.PCS_SKIP_REG3_9_8(0),
.PCS_SKIP_REG1_7_0(0),
.PCS_SKIP_REG2_7_0(0),
.PCS_SKIP_REG3_7_0(0),
.PCS_CFG_PRBS_ERR_O_SEL(0),
.PCS_CFG_PD_DELAY_RX(0),
.PCS_WR_START_GAP(0),
.PCS_MIN_IFG(0),
.PCS_INT_RX_MASK_0("FALSE"),
.PCS_INT_RX_MASK_1("FALSE"),
.PCS_INT_RX_MASK_2("FALSE"),
.PCS_INT_RX_MASK_3("FALSE"),
.PCS_INT_RX_MASK_4("FALSE"),
.PCS_INT_RX_MASK_5("FALSE"),
.PCS_INT_RX_CLR_5("FALSE"),
.PCS_INT_RX_CLR_4("FALSE"),
.PCS_INT_RX_CLR_3("FALSE"),
.PCS_INT_RX_CLR_2("FALSE"),
.PCS_INT_RX_CLR_1("FALSE"),
.PCS_INT_RX_CLR_0("FALSE"),
.PCS_EM_CNT_RD_EN("FALSE"),
.PCS_EM_CTRL_SEL("SIGNAL_CTRL"),
.PCS_EM_MODE_CTRL("HOLD"),
.PCS_EM_RD_CONDITION("TRIGGER"),
.PCS_EM_SP_PATTERN_7_0(0),
.PCS_EM_SP_PATTERN_15_8(0),
.PCS_EM_SP_PATTERN_23_16(0),
.PCS_EM_SP_PATTERN_31_24(0),
.PCS_EM_SP_PATTERN_39_32(0),
.PCS_EM_SP_PATTERN_47_40(0),
.PCS_EM_SP_PATTERN_55_48(0),
.PCS_EM_SP_PATTERN_63_56(0),
.PCS_EM_SP_PATTERN_71_64(0),
.PCS_EM_SP_PATTERN_79_72(0),
.PCS_EM_PMA_MASK_7_0(0),
.PCS_EM_PMA_MASK_15_8(0),
.PCS_EM_PMA_MASK_23_16(0),
.PCS_EM_PMA_MASK_31_24(0),
.PCS_EM_PMA_MASK_39_32(0),
.PCS_EM_PMA_MASK_47_40(0),
.PCS_EM_PMA_MASK_55_48(0),
.PCS_EM_PMA_MASK_63_56(0),
.PCS_EM_PMA_MASK_71_64(0),
```

```
.PCS_EM_PMA_MASK_79_72(0),
.PCS_EM_EYED_MASK_7_0(0),
.PCS_EM_EYED_MASK_15_8(0),
.PCS_EM_EYED_MASK_23_16(0),
.PCS_EM_EYED_MASK_31_24(0),
.PCS_EM_EYED_MASK_39_32(0),
.PCS_EM_EYED_MASK_47_40(0),
.PCS_EM_EYED_MASK_55_48(0),
.PCS_EM_EYED_MASK_63_56(0),
.PCS_EM_EYED_MASK_71_64(0),
.PCS_EM_EYED_MASK_79_72(0),
.PCS_EM_PRESCALE(0),
.PCS_CFG_TEST_STATUS_SEL("SEL_PMA_TEST_STATUS_INT"),
.PCS_CFG_DIFF_CNT_BND_RX(0),
.PCS_CFG_FLT_SEL_RX("FALSE"),
.PCS_FILTER_BND_RX(0),
.PCS_TCLK2FABRIC_DIV_RST_M("FALSE"),
.PCS_TX_PMA_TCLK_POLINV("PMA_TCLK"),
.PCS_TX_TCLK_POLINV("TCLK"),
.PCS_PCS_TCLK_SEL("PMA_TCLK"),
.PCS_GEAR_TCLK_SEL("PMA_TCLK"),
.PCS_TX_BRIDGE_TCLK_SEL("TCLK"),
.PCS_TCLK2ALIGNER_SEL("PMA_TCLK"),
.CA_DYN_DLY_EN_TX("FALSE"),
.PCS_TX_PCS_CLK_EN_SEL("HARDWIRED1"),
.PCS_TX_GEAR_CLK_EN_SEL("HARDWIRED0"),
.PCS_TCLK2FABRIC_DIV_EN("FALSE"),
.PCS_TCLK2FABRIC_SEL("CLK2ALIGNER_N_DIV2"),
.DLY_ADJUST_SIZE_TX(0),
.PCS_TX_PCS_TX_RSTN("FALSE"),
.PCS_TX_CA_RSTN("FALSE"),
.PCS_TX_SLAVE("MASTER"),
.PCS_TX_CA(0),
.PCS_CFG_PI_CLK_SEL(0),
.PCS_CFG_PI_CLK_EN_SEL("CLK_EN_ALWAYS1"),
.PCS_CFG_PI_STEP_SIZE_TX(0),
.PCS_CFG_SUM_THRESHOLD_TX(0),
.PCS_CFG_AVG_CYCLES_TX(0),
.PCS_CFG_NEGEDGE_EN_TX("FALSE"),
.PCS_CFG_ALIGN_THRD_TX(0),
.PCS_CFG_SCAN_INTERVAL_TX(0),
.PCS_CFG_STEP_SIZE_TX(0),
.PCS_CFG_REV_CNT_LIMIT_TX(0),
.PCS_CFG_FILTER_CNT_SIZE_TX(0),
.PCS_CFG_PI_DEFAULT_TX(0),
.PCS_CFG_PHDET_EN_TX("FALSE"),
.PCS_PMA_TX2RX_PLOOP_EN("FALSE"),
.PCS_PMA_TX2RX_SLOOP_EN("FALSE"),
```

```
.PCS_CFG_DYN_DLY_SEL_TX("FALSE"),
.PCS_CFG_DLY_REC_SIZE_TX(0),
.PCS_TX_DATA_WIDTH_MODE("8BIT"),
.PCS_TX_BYPASS_BRIDGE_UINT("FALSE"),
.PCS_TX_BYPASS_BRIDGE_FIFO("FALSE"),
.PCS_TX_BYPASS_GEAR("FALSE"),
.PCS_TX_BYPASS_ENC("FALSE"),
.PCS_TX_BYPASS_BIT_SLIP("FALSE"),
.PCS_TX_BRIDGE_GEAR_SEL("FALSE"),
.PCS_TXBRG_PWR_REDUCTION("NORMAL"),
.PCS_TXGEAR_PWR_REDUCTION("NORMAL"),
.PCS_TXENC_PWR_REDUCTION("NORMAL"),
.PCS_TXBSLP_PWR_REDUCTION("NORMAL"),
.PCS_TXPRBS_PWR_REDUCTION("NORMAL"),
.PCS_TXBRG_FULL_CHK_EN("FALSE"),
.PCS_TXBRG_EMPTY_CHK_EN("FALSE"),
.PCS_TX_ENCODER_MODE("DUAL_8B10B"),
.PCS_TX_PRBS_MODE("DISABLE"),
.PCS_TX_DRIVE_REG_MODE("NO_CHANGE"),
.PCS_TX_BIT_SLIP_CYCLES(0),
.PCS_TX_BASE_ADV_MODE("BASE"),
.PCS_TX_GEAR_SPLIT("NO_SPILT"),
.PCS_RX_BRIDGE_CLK_POLINV("N_CLK_INVERT"),
.PCS_PRBS_ERR_LPBK("FALSE"),
.PCS_TX_INSERT_ER("FALSE"),
.PCS_ENABLE_PRBS_GEN("FALSE"),
.PCS_FAR_LOOP("FALSE"),
.PCS_CFG_ENC_TYPE_EN("FALSE"),
.PCS_TXBRG_WADDR_START(0),
.PCS_TXBRG_RADDR_START(0),
.PCS_CFG_TX_PIC_EN("DISABLE"),
.PCS_CFG_PIC_DIRECT_INV("FALSE"),
.PCS_CFG_PI_MOD_CLK_EN("FALSE"),
.PCS_CFG_TX_MODULATOR_OW_EN("FALSE"),
.PCS_CFG_TX_PI_SSC_MODE_EN("FALSE"),
.PCS_CFG_TX_PI_OFFSET_MODE_EN("FALSE"),
.PCS_CFG_TX_PI_SSC_MODE_SEL(0),
.PCS_CFG_TXDEEMPH_EN("FALSE"),
.PCS_PI_STROBE_SEL("FALSE"),
.PCS_CFG_TX_PIC_GREY_SEL("FALSE"),
.PCS_CFG_PIC_RENEW_INV("NORMAL"),
.PCS_CFG_NUM_PIC(0),
.PCS_CFG_TXPIC_OW_EN("FALSE"),
.PCS_CFG_TX_SSC_OW_VALUE_0_7(0),
.PCS_INT_TX_MASK_0("FALSE"),
.PCS_INT_TX_MASK_1("FALSE"),
.PCS_INT_TX_MASK_2("FALSE"),
.PCS_TX_WPTR_SEL("FALSE"),
```

```
.PCS_INT_TX_CLR_2("FALSE"),
.PCS_INT_TX_CLR_1("FALSE"),
.PCS_INT_TX_CLR_0("FALSE"),
.PCS_CFG_PD_DELAY_TX(0),
.PCS_CFG_DIFF_CNT_BND_TX(0),
.PCS_CFG_PD_CLK_FR_CORE_SEL("FALSE"),
.PCS_CFG_FLT_SEL_TX("FALSE"),
.PCS_FILTER_BND_TX(0),
.PCS_CFG_TX_SSC_RANGE_7_0(0),
.PCS_CFG_TX_SSC_MODULATOR_EN("FALSE"),
.PCS_CFG_TX_SSC_SCALE2_SEL(0),
.PCS_CFG_TX_SSC_SCALE_SEL(0),
.PCS_CFG_TX_SSC_RANGE_8_9(0),
.PCS_CFG_TX_SSC_MODULATION_STEP_7_0(0),
.PCS_CFG_TX_SSC_OFFSET_7_0(0),
.PCS_CFG_TX_SSC_MODULATION_STEP_8(0),
.PCS_CFG_TX_SSC_OFFSET_8_9(0),
.PMA_REG_CHL_BIAS_POWER_SEL("FALSE"),
.PMA_REG_CHL_BIAS_POWER("FALSE"),
.PMA_REG_RX_BUSWIDTH("40BIT"),
.PMA_REG_RX_RATE("DIV4"),
.PMA_REG_RX_RATE_EN("FALSE"),
.PMA_REG_RX_RES_TRIM(55),
.PMA_REG_RX_SIGDET_STATUS_EN("FALSE"),
.PMA_REG_CDR_READY_THD_7_0(32),
.PMA_REG_CDR_READY_THD_11_8(0),
.PMA_REG_RX_BUSWIDTH_EN("FALSE"),
.PMA_REG_RX_PCLK_EDGE_SEL("POS_EDGE"),
.PMA_REG_RX_PIBUF_IC(3),
.PMA_REG_RX_DCC_IC_RX(1),
.PMA_REG_CDR_READY_CHECK_CTRL(0),
.PMA_REG_RX_ICTRL_TRX("100PCT"),
.PMA_REG_PRBS_CHK_WIDTH_SEL(1),
.PMA_REG_RX_ICTRL_PIBUF("100PCT"),
.PMA_REG_RX_ICTRL_PI("100PCT"),
.PMA_REG_RX_ICTRL_DCC("100PCT"),
.PMA_REG_TX_RATE("DIV1"),
.PMA_REG_TX_RATE_EN("FALSE"),
.PMA_REG_RX_TX2RX_PLPBK_RST_N("TRUE"),
.PMA_REG_RX_TX2RX_PLPBK_RST_N_EN("FALSE"),
.PMA_REG_RX_TX2RX_PLPBK_EN("FALSE"),
.PMA_REG_RX_DATA_POLARITY("NORMAL"),
.PMA_REG_RX_ERR_INSERT("FALSE"),
.PMA_REG_UDP_CHK_EN("FALSE"),
.PMA_REG_PRBS_SEL("PRBS7"),
.PMA_REG_PRBS_CHK_EN("FALSE"),
.PMA_REG_LPLL_NFC_STIC_DIS_N(0),
.PMA_REG_BIST_CHK_PAT_SEL("PRBS"),
```

```
.PMA_REG_LOAD_ERR_CNT("FALSE"),
.PMA_REG_CHK_COUNTER_EN("TRUE"),
.PMA_REG_CDR_PROP_GAN_SEL(3),
.PMA_REG_CDR_TUBO_PROP_GAIN_SEL(6),
.PMA_REG_CDR_INT_GAIN_SEL(2),
.PMA_REG_CDR_TUBO_INT_GAIN_SEL(6),
.PMA_REG_CDR_INT_SAT_MAX_4_0(0),
.PMA_REG_CDR_INT_SAT_MAX_9_5(28),
.PMA_REG_CDR_INT_SAT_MIN_2_0(0),
.PMA_REG_CDR_INT_SAT_MIN_9_3(16),
.PMA_ANA_RX_REG_O_61_55(21),
.PMA_ANA_RX_REG_O_69_62(0),
.PMA_ANA_RX_REG_O_77_70(135),
.PMA_ANA_RX_REG_O_85_78(1),
.PMA_ANA_RX_REG_O_93_86(8),
.PMA_ANA_RX_REG_O_100_94(64),
.PMA_ANA_RX_REG_O_108_101(0),
.PMA_ANA_RX_REG_O_111_109(0),
.PMA_REG_OOB_COMWAKE_GAP_MIN_4_0(3),
.PMA_REG_OOB_COMWAKE_GAP_MIN_5(0),
.PMA_REG_OOB_COMWAKE_GAP_MAX(11),
.PMA_REG_OOB_COMINIT_GAP_MIN(15),
.PMA_REG_OOB_COMINIT_GAP_MAX(35),
.PMA_REG_COMWAKE_STATUS_CLEAR(0),
.PMA_REG_COMINIT_STATUS_CLEAR(0),
.PMA_REG_RX_SATA_COMINIT_OW("FALSE"),
.PMA_REG_RX_SATA_COMINIT("FALSE"),
.PMA_REG_RX_SATA_COMWAKE_OW("FALSE"),
.PMA_REG_RX_SATA_COMWAKE("FALSE"),
.PMA_REG_RX_DCC_DISABLE("FALSE"),
.PMA_REG_RX_SLIP_SEL_EN("FALSE"),
.PMA_REG_RX_SLIP_SEL_3_0(0),
.PMA_REG_RX_SLIP_EN("FALSE"),
.PMA_REG_RX_SIGDET_STATUS_SEL(5),
.PMA_REG_RX_SIGDET_FSM_RST_N("TRUE"),
.PMA_REG_RX_SIGDET_STATUS("FALSE"),
.PMA_REG_RX_SIGDET_VTH("36MV"),
.PMA_REG_RX_SIGDET_GRM(0),
.PMA_REG_RX_SIGDET_PULSE_EXT("FALSE"),
.PMA_REG_RX_SIGDET_CH2_SEL(0),
.PMA_REG_RX_SIGDET_CH2_CHK_WINDOW(3),
.PMA_REG_RX_SIGDET_CHK_WINDOW_EN("TRUE"),
.PMA_REG_RX_SIGDET_NOSIG_COUNT_SETTING(4),
.PMA_REG_RX_SIGDET_OOB_DET_COUNT_VAL_2_0(0),
.PMA_REG_RX_SIGDET_OOB_DET_COUNT_VAL_4_3(0),
.PMA_REG_RX_SIGDET_4OOB_DET_SEL(7),
.PMA_REG_RX_SIGDET_IC_I(10),
.PMA_REG_RX_EQ1_R_SET_TOP(3),
```

```
.PMA_REG_RX_EQ1_C_SET_FB(0),
.PMA_REG_RX_EQ1_OFF("FALSE"),
.PMA_REG_RX_EQ2_R_SET_TOP(3),
.PMA_REG_RX_EQ2_C_SET_FB(0),
.PMA_REG_RX_EQ2_OFF("FALSE"),
.PMA_REG_RX_ICTRL_EQ(2),
.PMA_REG_EQ_DC_CALIB_EN("FALSE"),
.PMA_CTLE_CTRL_REG_I(0),
.PMA_CTLE_REG_FORCE_SEL_I("FALSE"),
.PMA_CTLE_REG_HOLD_I("FALSE"),
.PMA_CTLE_REG_INIT_DAC_I_1_0(0),
.PMA_CTLE_REG_INIT_DAC_I_3_2(0),
.PMA_CTLE_REG_POLARITY_I("FALSE"),
.PMA_CTLE_REG_SHIFTER_GAIN_I(4),
.PMA_CTLE_REG_THRESHOLD_I_1_0(0),
.PMA_CTLE_REG_THRESHOLD_I_9_2(1),
.PMA_CTLE_REG_THRESHOLD_I_11_10(0),
.PMA_REG_RX_RES_TRIM_EN("FALSE"),
.PMA_REG_ALG_RX_TERM_POWER_DIVIDING_SELECTION(1),
.PMA_REG_ALG_RX_TERM_VCM_SELECTION(3),
.PMA_REG_ALG_RX_TERM_TEST_SELECTION_7_0(0),
.PMA_REG_ALG_RX_TERM_TEST_SELECTION_9_8(0),
.PMA_REG_ALG_LOW_SPEED_MODE_ENABLE("FALSE"),
.PMA_REG_ALG_RX_CLOCK_POWER_DOWN_REGISTER("TRUE"),
.PMA_REG_ALG_RX_CLOCK_POWER_DOWN_SELECTION("FALSE"),
.PMA_REG_ALG_RX_DFE_POWER_DOWN_REGISTER_0("TRUE"),
.PMA_REG_ALG_RX_DFE_POWER_DOWN_SELECTION_1("FALSE"),
.PMA_REG_ALG_RX_CTLE_POWER_DOWN_REGISTER_0("TRUE"),
.PMA_REG_ALG_RX_CTLE_POWER_DOWN_SELECTION_1("FALSE"),
.PMA_REG_EYE_DFETAP1_PLORITY("FALSE"),
.PMA_REG_CDR_SEL("FALSE"),
.PMA_REG_EYE_DET_EN("FALSE"),
.PMA_REG_PI_BIAS_CURRENT(0),
.PMA_REG_ALG_DFE_TEST_SEL_6_0(0),
.PMA_REG_ALG_DFE_TEST_SEL_14_7(0),
.PMA_REG_ALG_DFE_TEST_SEL_21_15(0),
.PMA_REG_ALG_EYE_PATH_SEL(0),
.PMA_REG_ALG_CTLE_TEST_SEL(0),
.PMA_REG_RX_SLIP_SEL_4("FALSE"),
.PMA_REG_ALG_RX_T1_BUFF_EN("TRUE"),
.PMA_REG_ALG_RX_CDRX_BUFF_EN("TRUE"),
.PMA_REG_ALG_RX_VP_T1_SW_PLORITY("FALSE"),
.PMA_REG_ALG_RX_VP_PLORITY("TRUE"),
.PMA_REG_ALG_RX_GAIN_CTRL_SUMMER("FALSE"),
.PMA_REG_ALG_RX_DC_OFFSET_T1_EN("TRUE"),
.PMA_REG_ALG_RX_DC_OFFSET_VP_EN("TRUE"),
.PMA_REG_ALG_RX_DC_OFFSET_CDRX_EN("TRUE"),
.PMA_REG_ALG_RX_DC_OFFSET_CDRY_EN("TRUE"),
```

```
.PMA_REG_ALG_RX_DC_OFFSET_EYE_EN("TRUE"),
.PMA_REG_ALG_SLICER_DC_OFFSET_OVERWITE("FALSE"),
.PMA_REG_ALG_SLICER_DC_OFFSET_REG("FALSE"),
.PMA_REG_RX_PGA_OFF("FALSE"),
.PMA_REG_ALG_CDR_XWEIGHT_I(4),
.PMA_REG_ALG_CDR_YWEIGHT_I(4),
.PMA_REG_ALG_CTLE_FLIPDIR_I("FALSE"),
.PMA_REG_ALG_CTLE_HOLD_I("FALSE"),
.PMA_REG_ALG_CTLE_INITDAC_5_0(0),
.PMA_REG_ALG_CTLE_INITDAC_6(0),
.PMA_REG_ALG_CTLE_OVERWREN_I("FALSE"),
.PMA_REG_ALG_CTLE_SHIFT_I(4),
.PMA_REG_ALG_CTLE_TOPNUM_2_0(4),
.PMA_REG_ALG_CTLE_TOPNUM_4_3(2),
.PMA_REG_ALG_CTLEOFS_FLIPDIR_I("TRUE"),
.PMA_REG_ALG_CTLEOFS_HOLD_I("FALSE"),
.PMA_REG_ALG_CTLEOFS_INITDAC_3_0(0),
.PMA_REG_ALG_CTLEOFS_INITDAC_6_4(4),
.PMA_REG_ALG_CTLEOFS_OVERWREN("FALSE"),
.PMA_REG_ALG_CTLEOFS_SHIFT_I(4),
.PMA_REG_ALG_DFE_CTLE_PWD("FALSE"),
.PMA_REG_ALG_H1_FLIPDIR_I("FALSE"),
.PMA_REG_ALG_H1_HOLD_I("FALSE"),
.PMA_REG_ALG_H1_INITDAC_5_0(0),
.PMA_REG_ALG_H1_INITDAC_6(0),
.PMA_REG_ALG_H1_OVERWREN_I("FALSE"),
.PMA_REG_ALG_H1_SHIFT_I(4),
.PMA_REG_ALG_H2_FLIPDIR_I("FALSE"),
.PMA_REG_ALG_H2_HOLD_I("FALSE"),
.PMA_REG_ALG_H2_INITDAC_0(0),
.PMA_REG_ALG_H2_INITDAC_5_1(0),
.PMA_REG_ALG_H2_OVERWREN_I("FALSE"),
.PMA_REG_ALG_H2_SHIFT_I_1_0(0),
.PMA_REG_ALG_H2_SHIFT_I_2(1),
.PMA_REG_ALG_H3_FLIPDIR_I("FALSE"),
.PMA_REG_ALG_H3_HOLD_I("FALSE"),
.PMA_REG_ALG_H3_INITDAC_4_0(0),
.PMA_REG_ALG_H3_INITDAC_5(1),
.PMA_REG_ALG_H3_OVERWREN_I("FALSE"),
.PMA_REG_ALG_H3_SHIFT_I(4),
.PMA_REG_ALG_H4_FLIPDIR_I("FALSE"),
.PMA_REG_ALG_H4_HOLD_I("FALSE"),
.PMA_REG_ALG_H4_INITDAC_0(0),
.PMA_REG_ALG_H4_INITDAC_4_1(8),
.PMA_REG_ALG_H4_OVERWREN_I("FALSE"),
.PMA_REG_ALG_H4_SHIFT_I(4),
.PMA_REG_ALG_H5_FLIPDIR_I("FALSE"),
.PMA_REG_ALG_H5_HOLD_I("FALSE"),
```

```
.PMA_REG_ALG_H5_INITDAC(16),
.PMA_REG_ALG_H5_OVERWREN_I("FALSE"),
.PMA_REG_ALG_H5_SHIFT_I(4),
.PMA_REG_ALG_H6_FLIPDIR_I("FALSE"),
.PMA_REG_ALG_H6_HOLD_I("FALSE"),
.PMA_REG_ALG_H6_INITDAC_2_0(0),
.PMA_REG_ALG_H6_INITDAC_4_3(2),
.PMA_REG_ALG_H6_OVERWREN_I("FALSE"),
.PMA_REG_ALG_H6_SHIFT_I(4),
.PMA_REG_ALG_HCTLE_OFS_1_0(0),
.PMA_REG_ALG_HCTLE_OFS_3_2(2),
.PMA_REG_ALG_HCTLE_OVERWRDAC_5_0(0),
.PMA_REG_ALG_HCTLE_OVERWRDAC_6(1),
.PMA_REG_ALG_HCTLE_OVERWREN("FALSE"),
.PMA_REG_ALG_LEQH_INITDAC_I(0),
.PMA_REG_ALG_LEQH_PWD_I("FALSE"),
.PMA_REG_ALG_LEQH_REG_HOLD_I("FALSE"),
.PMA_REG_ALG_LEQH_REG_SHIFT_I(4),
.PMA_REG_ALG_LEQL_INITDAC_I(0),
.PMA_REG_ALG_LEQL_PWD_I("FALSE"),
.PMA_REG_ALG_LEQL_REG_HOLD_I("FALSE"),
.PMA_REG_ALG_LEQL_REG_PRESELECT_I(7),
.PMA_REG_ALG_LEQL_REG_SHIFT_I(4),
.PMA_REG_ALG_NEXTBIT_I("FALSE"),
.PMA_REG_ALG_SOFS_COUNTMAX_I_6_0(127),
.PMA_REG_ALG_SOFS_COUNTMAX_I_14_7(255),
.PMA_REG_ALG_SOFS_COUNTMAX_I_19_15(31),
.PMA_REG_ALG_SOFS_DACWIN_I(1),
.PMA_REG_ALG_SOFS_FLIP_DIR_I("TRUE"),
.PMA_REG_ALG_SOFS_FORCE_I("FALSE"),
.PMA_REG_ALG_SOFS_FORCEDAC_I_5_0(0),
.PMA_REG_ALG_SOFS_FORCEDAC_I_6(1),
.PMA_REG_ALG_SOFS_FORCENUM_I(0),
.PMA_REG_ALG_SOFS_INITDAC_2_0(0),
.PMA_REG_ALG_SOFS_INITDAC_6_3(8),
.PMA_REG_ALG_SOFS_SHIFT_I(1),
.PMA_REG_ALG_SOFS_SKIP_I("FALSE"),
.PMA_REG_ALG_SOFS_WINCOUNTMAX_I_7_0(254),
.PMA_REG_ALG_SOFS_WINCOUNTMAX_I_11_8(15),
.PMA_REG_ALG_ST_FLIPDIR_I("FALSE"),
.PMA_REG_ALG_ST_FORCEN("FALSE"),
.PMA_REG_ALG_ST_HOLD_I("FALSE"),
.PMA_REG_ALG_ST_INITDAC_0(0),
.PMA_REG_ALG_ST_INITDAC_4_1(8),
.PMA_REG_ALG_ST_RECALEN("FALSE"),
.PMA_REG_ALG_ST_SHIFT_I(4),
.PMA_REG_ALG_ST_STARTCNT_7_0(0),
.PMA_REG_ALG_ST_STARTCNT_15_8(128),
```

```
.PMA_REG_ALG_ST_STARTCNT_19_16(0),
.PMA_REG_ALG_ST_TAPCNT_3_0(0),
.PMA_REG_ALG_ST_TAPCNT_11_4(0),
.PMA_REG_ALG_ST_TAPCNT_17_12(2),
.PMA_REG_ALG_ST_TOPTAP_1_0(3),
.PMA_REG_ALG_ST_TOPTAP_3_2(3),
.PMA_REG_ALG_SWCLK_DIV(0),
.PMA_REG_ALG_TAPA_DAC_3_0(0),
.PMA_REG_ALG_TAPA_DAC_4(1),
.PMA_REG_ALG_TAPA_NUM(7),
.PMA_REG_ALG_TAPB_DAC_0(0),
.PMA_REG_ALG_TAPB_DAC_4_1(8),
.PMA_REG_ALG_TAPB_NUM_3_0(8),
.PMA_REG_ALG_TAPB_NUM_5_4(0),
.PMA_REG_ALG_TAPC_DAC(16),
.PMA_REG_ALG_TAPC_NUM_0(1),
.PMA_REG_ALG_TAPC_NUM_5_1(4),
.PMA_REG_ALG_TAPD_DAC_2_0(0),
.PMA_REG_ALG_TAPD_DAC_4_3(2),
.PMA_REG_ALG_TAPD_NUM(10),
.PMA_REG_ALG_VP_FLIPDIR_I("FALSE"),
.PMA_REG_ALG_VP_GRN_SHIFT_I(5),
.PMA_REG_ALG_VP_HOLD_I("FALSE"),
.PMA_REG_ALG_VP_IDEAL_2_0(0),
.PMA_REG_ALG_VP_IDEAL_6_3(10),
.PMA_REG_ALG_VP_INITDAC_I_3_0(0),
.PMA_REG_ALG_VP_INITDAC_I_6_4(0),
.PMA_REG_ALG_VP_OVERWREN("FALSE"),
.PMA_REG_ALG_VP_RED_SHIFT_I(5),
.PMA_REG_ALG_VPOFS_SEL_0(0),
.PMA_REG_ALG_VPOFS_SEL_2_1(1),
.PMA_REG_ALG_H1_UPBOUND_5_0(55),
.PMA_REG_ALG_H1_UPBOUND_6(1),
.PMA_REG_ALG_CTLEOFS_PWDN("FALSE"),
.PMA_REG_ALG_LEQH_OVEREN_I("FALSE"),
.PMA_REG_ALG_LEQL_OVEREN_I("FALSE"),
.PMA_REG_ALG_AGC_FLIPDIR_I("FALSE"),
.PMA_REG_ALG_AGC_HOLD_I("FALSE"),
.PMA_REG_ALG_AGC_INITDAC(10),
.PMA_REG_ALG_AGC_LOWBOUND_1_0(3),
.PMA_REG_ALG_AGC_LOWBOUND_3_2(0),
.PMA_REG_ALG_AGC_OVERWREN_I("FALSE"),
.PMA_REG_ALG_AGC_PWD("FALSE"),
.PMA_REG_ALG_AGC_SHIFT_I(4),
.PMA_REG_ALG_AGC_UPBOUND_0(1),
.PMA_REG_ALG_AGC_UPBOUND_3_1(7),
.PMA_REG_ALG_AGC_WAITSEL(11),
.PMA_REG_PI_CTRL_SEL_RX("FALSE"),
```

```
.PMA_REG_PI_CTRL_RX_4_0(0),
.PMA_REG_PI_CTRL_RX_7_5(0),
.PMA_CFG_RX_LANE_POWERUP("ON"),
.PMA_CFG_RX_PMA_RSTN("TRUE"),
.PMA_INT_PMA_RX_MASK_0("FALSE"),
.PMA_INT_PMA_RX_CLR_0("FALSE"),
.PMA_CFG_CTLE_ADP_RSTN("TRUE"),
.PMA_CFG_RX_CDR_RSTN("TRUE"),
.PMA_CFG_RX_CLKPATH_RSTN("TRUE"),
.PMA_CFG_RX_DFE_RSTN("TRUE"),
.PMA_CFG_RX_LEQ_RSTN("TRUE"),
.PMA_CFG_RX_SLIDING_RSTN("TRUE"),
.PMA_CFG_RX_EYE_RSTN("TRUE"),
.PMA_CFG_RX_CTLE_DCCAL_RSTN("TRUE"),
.PMA_CFG_RX_SLICER_DCCAL_RSTN("TRUE"),
.PMA_CFG_RX_SLIP_RSTN("TRUE"),
.PMA_REG_TX_BEACON_TIMER_SEL(0),
.PMA_REG_TX_BIT_CONV("FALSE"),
.PMA_REG_TX_RES_CAL(50),
.PMA_REG_TX_UDP_DATA_20(0),
.PMA_REG_TX_UDP_DATA_26_21(0),
.PMA_REG_TX_UDP_DATA_34_27(0),
.PMA_REG_TX_UDP_DATA_39_25(0),
.PMA_REG_TX_BUSWIDTH_EN(0),
.PMA_REG_TX_PD_POST("OFF"),
.PMA_REG_TX_PD_POST_OW("FALSE"),
.PMA_REG_TX_BUSWIDTH("20BIT"),
.PMA_REG_EI_PCLK_DELAY_SEL(0),
.PMA_REG_TX_AMP_DAC0(25),
.PMA_REG_TX_AMP_DAC1(19),
.PMA_REG_TX_AMP_DAC2(14),
.PMA_REG_TX_AMP_DAC3(9),
.PMA_REG_TX_RXDET_THRESHOLD("84MV"),
.PMA_REG_TX_BEACON_OSC_CTRL("FALSE"),
.PMA_REG_TX_PRBS_GEN_WIDTH_SEL(0),
.PMA_REG_TX_TX2RX_SLPBACK_EN("FALSE"),
.PMA_REG_TX_PCLK_EDGE_SEL("FALSE"),
.PMA_REG_TX_PRBS_GEN_EN("FALSE"),
.PMA_REG_TX_PRBS_SEL("PRBS7"),
.PMA_REG_TX_UDP_DATA_7_TO_0(5),
.PMA_REG_TX_UDP_DATA_15_TO_8(235),
.PMA_REG_TX_UDP_DATA_19_TO_16(3),
.PMA_REG_TX_FIFO_WP_CTRL(4),
.PMA_REG_TX_FIFO_EN("FALSE"),
.PMA_REG_TX_DATA_MUX_SEL(0),
.PMA_REG_TX_ERR_INSERT("FALSE"),
.PMA_REG_TX_SATA_EN("FALSE"),
.PMA_REG_RATE_CHANGE_TXPCLK_ON_OW("FALSE"),
```

```
.PMA_REG_RATE_CHANGE_TXPCLK_ON("TRUE"),
.PMA_REG_TX_CFG_POST1(0),
.PMA_REG_TX_CFG_POST2(0),
.PMA_REG_TX_OOB_DELAY_SEL(0),
.PMA_REG_TX_POLARITY("NORMAL"),
.PMA_REG_TX_LS_MODE_EN("FALSE"),
.PMA_REG_RX_JTAG_OE("TRUE"),
.PMA_REG_RX_ACJTAG_VHYSTSEL(0),
.PMA_REG_TX_RES_CAL_EN("FALSE"),
.PMA_REG_RX_TERM_MODE_CTRL(5),
.PMA_REG_PLPBK_TXPCLK_EN("FALSE"),
.PMA_REG_CLK_SEL_STROBE_TXPCLK(2),
.PMA_REG_TX_PH_SEL_0(1),
.PMA_REG_TX_PH_SEL_6_1(0),
.PMA_REG_TX_CFG_PRE(0),
.PMA_REG_TX_CFG_MAIN(0),
.PMA_REG_CFG_POST(0),
.PMA_REG_PD_MAIN("TRUE"),
.PMA_REG_PD_PRE("TRUE"),
.PMA_REG_TX_RXDET_TIMER_SEL(87),
.PMA_REG_TX_PI_CUR_BUF("10GHz"),
.PMA_REG_TX_MOD_STAND_BY_EN("FALSE"),
.PMA_REG_STATE_STAND_BY_SEL("FALSE"),
.PMA_REG_TX_SYNC_NEW("FALSE"),
.PMA_REG_TX_SYNC_NEW_OW("FALSE"),
.PMA_REG_TX_CHANGE_ON_OW("FALSE"),
.PMA_REG_TX_CHANGE_ON_POLAR_CTRL("FALSE"),
.PMA_REG_TX_FREERUN_PD("TRUE"),
.PMA_REG_TX_CHANGE_ON_SEL("FALSE"),
.PMA_REG_TX_CHANGE_ON_CTRL("FALSE"),
.PMA_REG_TX_FREERUN_RATE_0(0),
.PMA_REG_TX_FREERUN_RATE_1(0),
.PMA_REG_TX_FREERUN_RATE_OW("FALSE"),
.PMA_REG_TX_RST_SYNC_CLK_SEL("TRUE"),
.PMA_REG_TX_PI_CTRL_SEL(0),
.PMA_REG_TX_PI_CTRL(0),
.PMA_LANE_POWERUP("TRUE"),
.PMA_POR_N("TRUE"),
.PMA_TX_LANE_POWERUP("TRUE"),
.PMA_TX_PMA_RSTN("TRUE"),
.PMA_LPLL_POWERUP("TRUE"),
.PMA_LPLL_RSTN("TRUE"),
.PMA_LPLL_LOCKDET_RSTN("TRUE"),
.PMA_REG_LPLL_PFDDELAY_SEL(1),
.PMA_REG_LPLL_PFDDELAY_EN("TRUE"),
.PMA_REG_LPLL_VCTRL_SET(0),
.PMA_LPLL_CHARGE_PUMP_CTRL("type"),
.PMA_LPLL_REFDIV("DIV1"),
```

```
.PMA_LPLL_FBDIV(38),
.PMA_LPLL_LPF_RES(1),
.PMA_LPLL_REFLOSS_READY(0),
.PMA_LPLL_LOCKED_PFDDELAY(0),
.PMA_LPLL_MCLK_SEL(0),
.PMA_LPLL_TEST_SEL(0),
.PMA_LPLL_TEST_SIG_HALF_EN("TRUE"),
.PMA_LPLL_TEST_V_EN("FALSE"),
.PMA_LPLL_MCLK_EN("TRUE"),
.PMA_LPLL_MCLK_DET_CTL(16),
.PMA_LPLL_LOCKDET_REFCT(3),
.PMA_LPLL_LOCKDET_FBCT(3),
.PMA_LPLL_LOCKDET_LOCKCT(4),
.PMA_LPLL_LOCKDET_ITER(1),
.PMA_LPLL_UNLOCKDET_ITER(2),
.PMA_LPLL_LOCKDET_EN_OW(0),
.PMA_LPLL_LOCKDET_EN(0),
.PMA_LPLL_LOCKDET_MODE(0),
.PMA_LPLL_LOCKDET_OW(0),
.PMA_LPLL_LOCKDETED(0),
.PMA_LPLL_UNLOCKDET_OW(0),
.PMA_LPLL_UNLOCKDETED(0),
.PMA_LPLL_LOCKED_STICKY_CLEAR(0),
.PMA_LPLL_UNLOCKED_STICKY_CLEAR(0),
.PMA_LPLL_LOCKDET_REPEAT(0),
.PMA_LPLL_NOFBCLK_STICKY_CLEAR(0),
.PMA_LPLL_NOREFCLK_STICKY_CLEAR(0),
.PMA_LPLL_READY_OR_LOCK(0),
.PMA_LPLL_READY(0),
.PMA_LPLL_READY_OW(0),
.PMA_REG_TXCLK_SEL("HPLL"),
.PMA_REG_RXCLK_SEL("HPLL"),
.PMA_REG_TEST_BUF(0),
.PMA_REG_RX_DEF_SEL0("18.75uA"),
.PMA_REG_RX_DEF_SEL1("18.75uA"),
.PMA_REG_TX_DIV_SEL("25uA"),
.PMA_REG_LPLL_AMP_SEL("25uA"),
.PMA_REG_LPLL_VCO_SEL("25uA"),
.PMA_REG_LPLL_CHARGE_PUMP_SEL("25uA"),
.PMA_REG_RX_EQ0_SEL("56.25uA"),
.PMA_REG_RX_EQ1_SEL("56.25uA"),
.PMA_REG_RX_PGA_SEL("56.25uA"),
.PMA_REG_CHL_TEST(0),
.PMA_REG_IBIAS_STATIC_SEL_7_0(255),
.PMA_REG_IBIAS_STATIC_SEL_15_8(247),
.PMA_REG_IBIAS_STATIC_SEL_18_16(7),
.PMA_REG_IBIAS_STATIC_SEL_21_19(0),
.PMA_REG_IBIAS_STATIC_SEL_29_22(0),
```

```
.PMA_REG_IBIAS_STATIC_SEL_37_30(1),
.PMA_REG_IBIAS_DYNAMIC_PD_7_0(0),
.PMA_REG_IBIAS_DYNAMIC_PD_15_8(0),
.PMA_REG_IBIAS_DYNAMIC_PD_18_16(0),
.PMA_REG_IBIAS_STA_CUR_SEL_4_0(31),
.PMA_REG_IBIAS_STA_CUR_SEL_8_5(15),
.PMA_REG_IBIAS_STA_CUR_SEL_12_9(0),
.PMA_REG_IBIAS_STA_CUR_SEL_17_13(0),
.PMA_REG_IBIAS_STA_CUR_PD_2_0(0),
.PMA_REG_IBIAS_STA_CUR_PD_8_3(0),
.PMA_REG_BANDGAP_VOL_SEL("BANDGAP"),
.PMA_REG_BANDGAP_TEST(0)
) <InstanceName> (
.LANE_COUT_BUS_BACKWARD(),       // OUTPUT[1:0]
.LANE_COUT_BUS_FORWARD(),        // OUTPUT[24:0]
.P_CFG_RDATA(),                  // OUTPUT[7:0]
.P_EM_ERROR_CNT(),               // OUTPUT[2:0]
.P_RDATA(),                      // OUTPUT[87:0]
.P_RXSTATUS(),                   // OUTPUT[5:0]
.P_TEST_STATUS(),                // OUTPUT[19:0]
.LANE_CIN_BUS_BACKWARD(),        // INPUT[1:0]
.LANE_CIN_BUS_FORWARD(),         // INPUT[24:0]
.P_ALIGN_MODE_RX(),              // INPUT[1:0]
.P_ALIGN_MODE_TX(),              // INPUT[2:0]
.P_CFG_ADDR(),                   // INPUT[11:0]
.P_CFG_WDATA(),                  // INPUT[7:0]
.P_CIM_CLK_ALIGNER_RX(),         // INPUT[7:0]
.P_CIM_CLK_ALIGNER_TX(),         // INPUT[7:0]
.P_EM_MODE_CTRL(),               // INPUT[1:0]
.P_RX_BUSWIDTH(),                // INPUT[2:0]
.P_RX_EYE_TAP(),                 // INPUT[7:0]
.P_RX_PIC_EYE(),                 // INPUT[7:0]
.P_RX_PIC_FASTLOCK(),            // INPUT[7:0]
.P_RX_RATE(),                    // INPUT[1:0]
.P_TDATA(),                      // INPUT[87:0]
.P_TX_BUSWIDTH(),                // INPUT[2:0]
.P_TX_DEEMP(),                   // INPUT[15:0]
.P_TX_DEEMP_POST_SEL(),          // INPUT[1:0]
.P_TX_FREERUN_BUSWIDTH(),        // INPUT[2:0]
.P_TX_MARGIN(),                  // INPUT[2:0]
.P_TX_RATE(),                    // INPUT[1:0]
.P_CA_ALIGN_RX(),                // OUTPUT
.P_CA_ALIGN_TX(),                // OUTPUT
.P_CFG_INT(),                    // OUTPUT
.P_CFG_READY(),                  // OUTPUT
.P_LPLL_READY(),                 // OUTPUT
.P_PCS_LSM_SYNCED(),             // OUTPUT
.P_PCS_RX_MCB_STATUS(),          // OUTPUT
```

```
.P_RCLK2FABRIC(),            // OUTPUT
.P_RXDVLD(),                 // OUTPUT
.P_RXDVLD_H(),               // OUTPUT
.P_RX_LS_DATA(),             // OUTPUT
.P_RX_PRBS_ERROR(),          // OUTPUT
.P_RX_READY(),               // OUTPUT
.P_RX_SATA_COMINIT(),        // OUTPUT
.P_RX_SATA_COMWAKE(),        // OUTPUT
.P_RX_SIGDET_STATUS(),       // OUTPUT
.P_TCLK2FABRIC(),            // OUTPUT
.P_TX_RXDET_STATUS(),        // OUTPUT
.P_TX_SDN(),                 // OUTPUT
.P_TX_SDP(),                 // OUTPUT
.PMA_HPLL_CK0(),             // INPUT
.PMA_HPLL_CK90(),            // INPUT
.PMA_HPLL_CK180(),           // INPUT
.PMA_HPLL_CK270(),           // INPUT
.PMA_HPLL_READY_IN(),        // INPUT
.PMA_HPLL_REFCLK_IN(),       // INPUT
.PMA_TX_SYNC_HPLL_IN(),      // INPUT
.P_ALIGN_MODE_VALID_RX(),    // INPUT
.P_ALIGN_MODE_VALID_TX(),    // INPUT
.P_BLK_ALIGN_CTRL(),         // INPUT
.P_CFG_CLK(),                // INPUT
.P_CFG_ENABLE(),             // INPUT
.P_CFG_PSEL(),               // INPUT
.P_CFG_RST(),                // INPUT
.P_CFG_WRITE(),              // INPUT
.P_EM_RD_TRIGGER(),          // INPUT
.P_EXT_BRIDGE_PCS_RST(),     // INPUT
.P_LANE_POWERDOWN(),         // INPUT
.P_LANE_RST(),               // INPUT
.P_LPLL_LOCKDET_RST(),       // INPUT
.P_LPLL_POWERDOWN(),         // INPUT
.P_LPLL_REFCLK_IN(),         // INPUT
.P_LPLL_RST(),               // INPUT
.P_PCIE_EI_H(),              // INPUT
.P_PCIE_EI_L(),              // INPUT
.P_PCS_BIT_SLIP(),           // INPUT
.P_PCS_FAREND_LOOP(),        // INPUT
.P_PCS_MCB_EXT_EN(),         // INPUT
.P_PCS_NEAREND_LOOP(),       // INPUT
.P_PCS_PRBS_EN(),            // INPUT
.P_PCS_RX_RST(),             // INPUT
.P_PCS_TX_RST(),             // INPUT
.P_PCS_WORD_ALIGN_EN(),      // INPUT
.P_PMA_FAREND_PLOOP(),       // INPUT
.P_PMA_NEAREND_PLOOP(),      // INPUT
```

```
.P_PMA_NEAREND_SLOOP(),          // INPUT
.P_RCLK2_FR_CORE(),              // INPUT
.P_RX_CDRX_EN(),                 // INPUT
.P_RX_CDR_RST(),                 // INPUT
.P_RX_CLKPATH_RST(),             // INPUT
.P_RX_CLK_FR_CORE(),             // INPUT
.P_RX_CTLE_DCCAL_EN(),           // INPUT
.P_RX_CTLE_DCCAL_RST(),          // INPUT
.P_RX_DEC_TYPE(),                // INPUT
.P_RX_DFE_EN(),                  // INPUT
.P_RX_DFE_RST(),                 // INPUT
.P_RX_EYE_EN(),                  // INPUT
.P_RX_EYE_RST(),                 // INPUT
.P_RX_HIGHZ(),                   // INPUT
.P_RX_LANE_POWERDOWN(),          // INPUT
.P_RX_LEQ_RST(),                 // INPUT
.P_RX_PIC_FASTLOCK_STROBE(),     // INPUT
.P_RX_PMA_RST(),                 // INPUT
.P_RX_POLARITY_INVERT(),         // INPUT
.P_RX_SDN(),                     // INPUT
.P_RX_SDP(),                     // INPUT
.P_RX_SLICER_DCCAL_EN(),         // INPUT
.P_RX_SLICER_DCCAL_RST(),        // INPUT
.P_RX_SLIDING_EN(),              // INPUT
.P_RX_SLIDING_RST(),             // INPUT
.P_RX_SLIP_EN(),                 // INPUT
.P_RX_SLIP_RST(),                // INPUT
.P_RX_T1_DFE_EN(),               // INPUT
.P_RX_T1_EN(),                   // INPUT
.P_RX_T2_DFE_EN(),               // INPUT
.P_RX_T3_DFE_EN(),               // INPUT
.P_RX_T4_DFE_EN(),               // INPUT
.P_RX_T5_DFE_EN(),               // INPUT
.P_RX_T6_DFE_EN(),               // INPUT
.P_TCLK2_FR_CORE(),              // INPUT
.P_TX_BEACON_EN(),               // INPUT
.P_TX_CLK_FR_CORE(),             // INPUT
.P_TX_ENC_TYPE(),                // INPUT
.P_TX_LANE_POWERDOWN(),          // INPUT
.P_TX_LS_DATA(),                 // INPUT
.P_TX_PIC_EN(),                  // INPUT
.P_TX_PMA_RST(),                 // INPUT
.P_TX_RATE_CHANGE_ON_0(),        // INPUT
.P_TX_RATE_CHANGE_ON_1(),        // INPUT
.P_TX_RXDET_REQ(),               // INPUT
.P_TX_SWING(),                   // INPUT
.P_TX_SYNC()                     // INPUT
);
```

(8) GTP_HSSTHP_HPLL instruction

1. Supported devices

Table 352 GTP_HSSTHP_HPLL device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

HSSTHP_HPLL_TOP includes the following submodules:

1. A module in PCS: pcs13g_pcs_top_pll_single

2. A module in PMA: pma_common (with HPLL inside), which includes pma_qpll_top and pma_hpll_interface. HSSTHP_HPLL exists as a prim, and adds parameter to sc conversion on the basis of HSSTHP_HPLL_TOP to form GTP_HSSTHP_HPLL, so that HSSTHP_HPLL can be easily converted to GTP_HSSTHP_HPLL (meeting "GTP coding specification.doc"). HSSTHP_HPLL contains a HPLL (cooperating with LPLL to cover the line rate of 0.6GHz~13.125GHz). Users use this PLL in high-speed scenarios. GTP_HSSTHP_HPLL is a dedicated Hpll module for HSSTHP (high-performance high-speed serial transceiver). Series FPGA High-Speed Serial Transceiver (HSSTHP) User's Guide.

3. Port description

Table 353 GTP_HSSTHP_HPLL port list

Port Name	Direction	Data width	Functional description
P_CFG_RST_HPLL	I	1	Reset signal of the configuration interface, high effective. After reset, all registers are restored to the initial values set by Parameter
P_CFG_CLK_HPLL	I	1	Configure the clock input of the interface
P_CFG_PSEL_HPLL	I	1	Select input of configuration interface, active high
P_CFG_ENABLE_HPLL	I	1	Configure the access enable of the interface, high effective
P_CFG_WRITE_HPLL	I	1	Configure the read and write selection signal of the interface, high level means write, low level means read
P_CFG_ADDR_HPLL	I	12	Configure the address bus of the interface
P_CFG_WDATA_HPLL	I	8	Write data for configuration interface
P_CFG_READY_HPLL	O	1	Read and write ready output of dynamic configuration interface, from HPLL, high effective
P_CFG_RDATA_HPLL	O	8	Read data bus for configuration interface
P_CFG_INT_HPLL	O	1	Interrupt output of configuration interface, active high
P_HPLL_POWERDOWN	I	1	HPLL's powerdown control port
P_HPLL_RST	I	1	HPLL reset port
P_HPLL_LOCKDET_RST	I	1	hpll lock detector reset port
P_RES_CAL_RST	I	1	resistor calibration reset
P_TX_SYNC	I	1	global tx sync input; 0:normal work(default); 1: sync.

Port Name	Direction	Data width	Functional description
P_HPLL_DIV_SYNC	I	1	hpll output high-speed frequency divider synchronization signal 0: normal work (default); 1: sync.
P_REFCLK_DIV_SYNC	I	1	hpll reference clock divider sync signal 0:normal work(default); 1:sync.
P_HPLL_VCO_CALIB_EN	I	1	Hpll VCO calibration enable
P_RESCAL_I_CODE_I	I	6	Resistance correction code from fabric
P_RES_CAL_CODE_FABRIC	O	6	Resistance correction code sent to fabric
P_HPLL_READY	O	1	HPLL lock flag signal, sent to fabric
P_HPLL_DIV_CHANGE	I	1	enable pma config to change when pcie mode alter
PMA_HPLL_READY_O	O	1	HPLL lock flag, sent to lane0/1/2/3
PMA_HPLL_REFCLK_O	O	1	HPLL reference clock, sent to lane0/1/2/3
PMA_TX_SYNC_HPLL_O	O	1	Tx sync signal sent to lane0/1/2/3 after hpll
PMA_HPLL_CK0	O	1	The 0-degree output clock of hpll is sent to lane0/1/2/3
PMA_HPLL_CK90	O	1	The 90-degree output clock of hpll is sent to lane0/1/2/3
PMA_HPLL_CK180	O	1	The 180-degree output clock of hpll is sent to lane0/1/2/3
PMA_HPLL_CK270	O	1	The 270-degree output clock of hpll is sent to lane0/1/2/3
TX_SYNC_REFSYNC_O	O	1	The tx_sync signal after the reference clock synchronization is output to lane or hpll, or cascaded to other hsst
REFCLK_SYNC_REFSYNC_O	O	1	The refclk_sync signal after reference clock synchronization is output to lane or hpll, or cascaded to other hsst
DIV_SYNC_REFSYNC_O	O	1	The div_sync signal after reference clock synchronization is output to lane or hpll, or cascaded to other hsst
P_HPLL_REFCLK_I	I	1	hpll Reference clock input port, which can be connected to the REFCLK_OUTP port of bufds, or the clock input from srb (but the same hsst is required to have at most one reference clock input from srb)
REFCLK_TO_TX_SYNC_I	I	1	The reference clock required for synchronization of the tx_sync signal, connected to the REFCLK_OUTP port of bufds
REFCLK_TO_REFCLK_SYNC_I	I	1	The reference clock required for synchronization of the refclk_sync signal, connected to bufds The REFCLK_OUTP port
REFCLK_TO_DIV_SYNC_I	I	1	The reference clock required for synchronization of the div_sync signal, connected to the REFCLK_OUTP port of bufds
ANA_TX_SYNC_I	I	1	tx_sync signal input port, when the user needs to use the asynchronous tx_sync signal, this port is connected to the same signal as the P_TX_SYNC port; when the user needs to use the synchronous tx_sync signal, this port is connected to the TX_SYNC_REFSYNC_O port output by hpll
ANA_HPLL_REFCLK_SYNC_I	I	1	refclk_sync signal input port, when the user needs to use the asynchronous refclk_sync signal, this port is connected to the same signal as the P_REFCLK_DIV_SYNC port; when the user needs to use the synchronous refclk_sync signal, this port is connected to the REFCLK_SYNC_REF SYNC_O port output by hpll
ANA_HPLL_DIV_SYNC_I	I	1	div_sync signal input port, when the user needs to use the asynchronous div_sync signal, this port is connected to the same signal as the P_HPLL_DIV_SYNC port; when the user needs to use the synchronous div_sync signal, this port is connected to the DIV_SYNC_REFSYNC_O port output by hpll

Port Name	Direction	Data width	Functional description
P_PLL_REFCLK6_I	I	1	Added a new virtual port, through which the fabric reference clock is sent to the HSSTHP during automatic resistance calibration. This port does not support functional simulation, but only supports on-board testing.

Note: 1. DFT related ports are not included in the GTP file.

2. The ports cascaded across Quads have been deleted from GTP_HSSTHP_HPLL, and the PDS software will automatically map and cascade according to the connection relationship between GTP ports. Users do not need to pay attention to ports such as P_FROM_LOWER_HSST_BUS and P_FOR_LOWER_HSST_BUS.

4. Parameter description

Table 354 GTP_HSSTHP_HPLL parameter list

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
PMA_REG_REFCLK0_ IMPEDANCE_SEL	<string>	"100_OHM"; "HIGH_Z"	"100_OHM"	Reference clock0 input impedance selection; "HIGH_Z": high z; "100_OHM": 100ohm.
PMA_REG_REFCLK1_ IMPEDANCE_SEL	<string>	"100_OHM"; "HIGH_Z"	"100_OHM"	Reference clock1 input impedance selection; "HIGH_Z": high z; "100_OHM": 100ohm.
PMA_REG_PLL_JTAG0_ VTH_SEL	<string>	"60MV"; "120MV"; "180MV"; "240MV"	"60MV"	Threshold voltage setting for PLL JTAG0 detection; "60MV": 60mV; "120MV": 120mV; "180MV": 180mV; "240MV": 240mV.
PMA_REG_PLL_JTAG0_ LPF_RSEL	<string>	"20K"; "10K"	"20K"	PLL JTAG0 LPF Resister selection; "20K": R=20k; "10K": R=10K.
PMA_REG_PLL_JTAG1_ VTH_SEL	<string>	"60MV"; "120MV"; "180MV"; "240MV"	"60MV"	Threshold voltage setting for PLL JTAG1 detection; "60MV": 60mV; "120MV": 120mV; " 180MV": 180mV; "240MV": 240mV.
PMA_REG_PLL_JTAG1_ LPF_RSEL	<string>	"20K"; "10K"	"20K"	PLL JTAG1 LPF Resister selection; "20K": R=20k; "10K": R=10K.
PMA_REG_IBIAS_ DYNAMIC_PD_7_0	<integer>	[255, 0]	0	Corresponding register reg_ibup_dyn_pd<8:1>, ibias dynamic current power down control; 1: power down; 0: work
PMA_REG_IBIAS_ DYNAMIC_PD_15_8	<integer>	[255, 0]	0	Corresponding register reg_ibup_dyn_pd<16:9>, ibias dynamic current power down control; 1: power down; 0: work
PMA_REG_IBIAS_ DYNAMIC_PD_18_16	<integer>	[7, 0]	0	Corresponding to reg_ibup_dyn_pd<19:17> register, ibias dynamic current power down control; 1: power down; 0: work

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_IBIAS_STA_CUR_PD_2_0	<integer>	[7, 0]	0	Corresponding to reg_ibup_sta_pd<3:1> register, ibias static current power down control; 1: power down; 0: work
PMA_REG_IBIAS_STA_CUR_PD_8_3	<integer>	[63, 0]	0	Corresponding to reg_ibup_sta_pd<9:4> register, ibias static current power down control; 1: power down; 0: work
PMA_REG_BANDGAP_VOL_SEL	<string>	"BANDGAP"; "VDDA"	"BANDGAP"	bandgap voltage sel; "BANDGAP": bandgap; "VDDA": vdda
PMA_REG_BANDGAP_TEST	<integer>	[15, 0]	0	bandgap test
PMA_REG_CALIB_CLKDIV_RATION	<string>	"DIV1"; "DIV2"; "DIV4"; "DIV8"	"DIV1"	set resistor calibration clock dividing ration; "DIV1": div1; "DIV2": div2; "DIV4": div4; "DIV8": div8
PMA_REG_TX_RATE_CHANGE_SEL0	<string>	"CLK_FROM_HPLL"; "CLK_FROM_HPLL_REFERENCE"	"CLK_FROM_HPLL"	bandgap powerdown register configuration: "CLK_FROM_HPLL": work; "CLK_FROM_HPLL_REFERENCE":power down.
PMA_REG_TX_RATE_CHANGE_SEL1	<string>	"SEL_SYNC_RATE_CHANGE"; "SEL_SYNC_FROM_PORT"	"SEL_SYNC_RATE_CHANGE"	Bandgap module enable selection: 0: Select slave register configuration, from reg_tx_rate_change_sel0 register; 1: Select the control port from ana_com_pd_o
PMA_REG_TLING_IMPEDANCE_CTRL	<string>	"125OHM"; "HIGH_Z"; "500OHM"; "250OHM"; "100OHM"; "83OHM"; "70OHM"	"125OHM"	Tline impeadance control; "HIGH_Z": high z; "500OHM": 500ohm; "250OHM": 250ohm; "125OHM": 125ohm; "100OHM": 100ohm ; "83OHM": 83ohm; "70OHM": 70ohm.
PMA_ANA_COM_REG_ 143	<integer>	0; 1	0	Select the edge of the reference clock to synchronize the sync signal of the hpll reference clock divider, 0: select rising edge; 1: Select falling edge.
PMA_ANA_COM_REG_ 149	<integer>	0; 1	0	selects the edge of the reference clock to synchronize the hpll divider sync signal, 0: rising edge; 1: Falling edge.
PMA_ANA_COM_REG_155	<integer>	0; 1	0	Select the edge of the reference clock to synchronize the tx sync signal; 0: rising edge, 1: Falling edge.

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_HPLL_DIV_ CHANGE_3_0	<integer>	[15, 0]	0	When hpll_div_change_i is 1, hpll frequency division configuration comes from reg_hpll_div_chang (pma_com_reg_i[177:162])
PMA_REG_HPLL_DIV_ CHANGE_11_4	<integer>	[255, 0]	0	
PMA_REG_HPLL_DIV_ CHANGE_15_12	<integer>	[15, 0]	0	
PMA_REG_HPLL_ VCOLDO_EN	<string>	"TRUE"; "FALSE"	"TRUE"	vco ldo enable port: "TRUE": enable; "FALSE": disable
PMA_REG_HPLL_ REFCLK_DIV	<integer>	1; 2; 3; 4; 5	1	hpll reference clock divider ratio: M: refdiv<5:0> 1 : xxxxx1; 2 : 000100; 3 : 000110; 4 : 001000; 5 : 001010.
PMA_REG_HPLL_ CHARGE_PUMP_PD	<string>	"WORK"; "POWER_DOWN"	"WORK"	charge pump power down control: "POWER_DOWN": power down; "WORK": work
PMA_REG_HPLL_ CHARGE_PUMP_CTRL	<integer>	[15, 0]	10	Charg pump charge and discharge current unit switch control: 0: 0000; 1: 0001; 2: 0010; ...... 15: 1111.
PMA_REG_HPLL_LPF_ RES_SEL	<string>	"5K"; "55K"; "12.7K"; "7.17K"; "3.83K"	"5K"	lpf_res resistance setting: "55K": 55k; "12.7K": 12.7k; "7.17K": 7.17k; "5K": 5k; "3.83K": 3.83k.
PMA_REG_HPLL_VCO_EN	<integer>	0; 1	1	vco enable port: 1: enable; 0: disable
PMA_REG_HPLL_ PCURRENT_SEL1	<integer>	0; 1	0	N tube bias current adjustment: 0: 4*I0; 1: 6*I0
PMA_REG_REFCLK_ RST_SEL	<integer>	0; 1	0	Reference clock reset selection control signal: 0: Select hpll_rst as the reference clock reset signal (applied in single quad working mode); 1: Select ana_refclk_sync as the complex of the reference clock Bit signal (applied in multiple quad working mode).
PMA_REG_NFC_STIC_ DIS_N	<integer>	0; 1	0	When 1, pll_nofbclk_sticky_o does not affect pll_locked_o; When 0, pll_nofbclk_sticky_o is 1 pll_locked_o is 0

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_HPLL_FBDIV0	<integer>	[127,0]	20	The calculation formula of HPLL fbdiv is as follows: D2= pma_hpll_reg_i<50:44>, corresponding to this parameter, fbdiv=(D1+D2)*2 D1=up<6:0> D2=dn<6:0> eg:D1=up<6:0>=20 (decimal) D2=dn<6:0>=20 (decimal) fbdiv=(20+20)*2=80
PMA_REG_HPLL_FBDIV1	<integer>	[127,0]	20	The calculation formula of HPLL fbdiv is as follows: D1= pma_hpll_reg_i<57:51>, corresponding to this parameter, fbdiv=(D1+D2)*2 D1=up<6:0> D2=dn<6:0> eg: D1=up<6:0>=20 (decimal) D2=dn<6:0>=20 (decimal) fbdiv=(20+20)*2=80
PMA_REG_HPLL_ PHASE_SEL	<string>	"DIV2"; "DIV1P5"	"DIV2"	hpll output frequency division selection: "DIV2": div2; "DIV1P5": div1p5.
PMA_REG_HPLL_CFG_7_0	<integer>	[255, 0]	7	Corresponding register hpll_cfg<7:0>: hpll_cfg<0>---clkmux_l_en, refers to the enable port of the left channel of clkmux: 1: enable 0: disable hpll_cfg<1>---clkmux_l_en, refers to the enable end of the right channel of clkmux: 1: enable 0: disable hpll_cfg<2>---en_osc_mclk 0: disable 1: enable hpll_cfg<3>--- sel_osc_mclk 0: osc_mclk output 36MHz; 1: sel_osc_mclk output 18Mhz hpll_cfg<7:4>=vco_cal manual configuration, the default configuration is 0000
PMA_REG_HPLL_CFG_15_8	<integer>	[255, 0]	132	Corresponding register hpll_cfg<15:8>: hpll_cfg<8>=vco_cal manual configuration, default configuration 0 hpll_cfg<9>---tx sync signal selection 0: use fully synchronous tx sync signal 1: directly use the high-speed output of hpll for synchronization hpll_cfg<10>---hpll_sel_lo selection: 0: Select the default digital signal; 1: Select hpll_set_hi to invert the output; hpll_cfg<15:11> --- refclk_det_ct<4:0> refers to the osc_clk count period: 11111: 2^4 11110: 2^5 11100: 2^6 11000: 2^7 10000: 2^8 (default) 00000: 2^9

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_PLL_LOCKDET_ RESET_N	<string>	"FALSE"; "TRUE"	"FALSE"	pll_lockdet_reset_n
PMA_REG_PLL_LOCKDET_ RESET_N_OW	<string>	"FALSE"; "TRUE"	"FALSE"	pll_lockdet_reset_n overwrite
PMA_REG_READY_ OR_LOCK	<string>	"FALSE"; "TRUE"	"FALSE"	Lock and ready selection; "FALSE": selection ready "TRUE": selection lock.
PMA_REG_HPLL_READY	<string>	"FALSE"; "TRUE"	"FALSE"	hpll_ready
PMA_REG_HPLL_ READY_OW	<string>	"TRUE"; "FALSE"	"FALSE"	hpll_ready overwrite control
PMA_REG_PLL_LOCKDET_ REFCT_0	<string>	"TRUE"; "FALSE"	"TRUE"	Together form the reference clock domain counter overflow value: Counter value for reference clock time out value of lock detect 000: 2^9 001: 2^10 010: 2^11 011: 2^12 100: 2^13 101: 2^14 110: 2^15 111: 2^16-1; The previous parameter "TRUE"=1, "FALSE"=0.
PMA_REG_PLL_LOCKDET_ REFCT_2_1	<integer>	[3, 0]	1	
PMA_REG_PLL_LOCKDET_ FBCT	<integer>	[7, 0]	3	Feedback clock domain counter overflow value: counter value for feedback clock time out value of lock detect 0: 2^9 1: 2^10 2: 2^11 3: 2^12 4: 2^13 5: 2^14 6: 2^15 7: 2^16-1
PMA_REG_PLL_LOCKDET_ LOCKCT	<integer>	[7, 0]	4	Difference counter overflow value Register value of lock detection threshold for count difference between ref_clk & fb_clk counters. 0: 2 1: 4 2: 8 3: 16 4: 32 (default) 5: 64 6: 128 7: 256

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_PLL_LOCKDET_ITER	<integer>	[7, 0]	0	lock cycle times Register value of number of iterations of consecutive successful one-time lock detection before set pll_lock. 0: 1 1: 2 2: 4 3: 8 (default) 4: 16 5: 32 6: 64 7: 127
PMA_REG_PLL_UNLOCKDET_ITER	<integer>	[3, 0]	2	Number of non-locking cycles Register value of number of iterations of lock detection attemps before set pll_unlock. 0: 64 1: 128 2: 256 (default) 3: 1023
PMA_REG_PLL_LOCKDET_EN_OW	<string>	"FALSE"; "TRUE"	"FALSE"	pll_lockdet_en overwrite
PMA_REG_PLL_LOCKDET_EN	<string>	"FALSE"; "TRUE"	"FALSE"	pll_lockdet_en
PMA_REG_PLL_LOCKDET_MODE	<string>	"FALSE"; "TRUE"	"FALSE"	Lockdet mode configuration; "FALSE": Mode 1; "TRUE": Mode 2.
PMA_REG_PLL_LOCKED_OW	<string>	"FALSE"; "TRUE"	"FALSE"	pll_locked_ow
PMA_REG_PLL_LOCKED	<string>	"FALSE"; "TRUE"	"FALSE"	PLL Locked Signal for Register Input
PMA_REG_PLL_UNLOCKED_OW	<string>	"FALSE"; "TRUE"	"FALSE"	pll_unlocked_ow
PMA_REG_PLL_UNLOCKED	<string>	"FALSE"; "TRUE"	"FALSE"	pll_unlocked
PMA_REG_PLL_LOCKED_STICKY_CLEAR	<string>	"FALSE"; "TRUE"	"FALSE"	Pll continuous lock signal clear signal Clear pll_locked_sticky_o when "TRUE"
PMA_REG_PLL_UNLOCKED_STICKY_CLEAR	<string>	"FALSE"; "TRUE"	"FALSE"	Pll continuous loss of lock signal clear signal Clear pll_unlocked_sticky_o when "TRUE"
PMA_REG_LOCKDET_REPEAT	<string>	"FALSE"; "TRUE"	"FALSE"	Retest after lock
PMA_REG_NOFBCLK_STICKY_CLEAR	<string>	"FALSE"; "TRUE"	"FALSE"	When the pll_nofbclk_sticky_o clear signal is "TRUE", pll_nofbclk_sticky_o is cleared to zero

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_NOREFCLK_STICKY_CLEAR	<string>	"FALSE"; "TRUE"	"FALSE"	pll_norefclk_sticky_o clear signal Clear pll_norefclk_sticky_o when "TRUE"
PMA_REG_RESCAL_EN	<string>	"FALSE"; "TRUE"	"FALSE"	Electric group calibration enable
PMA_REG_RESCAL_RST_N_OW	<string>	"FALSE"; "TRUE"	"FALSE"	rescal_rst_n_ow
PMA_REG_RESCAL_RST_N_VAL	<string>	"FALSE"; "TRUE"	"FALSE"	rescal_rst_n
PMA_REG_RESCAL_DONE_VAL	<string>	"FALSE"; "TRUE"	"FALSE"	Resistance calibration completed, reg value Register value to overwrite rescal_done output; "FALSE": resistor calibration is not done (default); "TRUE": resistor calibration is done.
PMA_REG_RESCAL_DONE_OW	<string>	"FALSE"; "TRUE"	"FALSE"	rescal_done_ow
PMA_REG_RESCAL_I_CODE_VAL_1_0	<integer>	[3, 0]	0	Together they form the average value of the variable resistor register control code
PMA_REG_RESCAL_I_CODE_VAL_5_2	<integer>	[15, 0]	0	
PMA_REG_RESCAL_I_CODE_OW	<string>	"FALSE"; "TRUE"	"FALSE"	rescal_i_code overwrite control
PMA_REG_RESCAL_ITER_VALID_SEL	<integer>	[3, 0]	0	iteration number of valid resistor calibration (valid means calibration result is in reasonable range) 0: 64 1: 32 2: 16 3: 8
PMA_REG_RESCAL_WAIT_SEL	<string>	"FALSE"; "TRUE"	"FALSE"	Wait times selection time interval of rescal_i_tune update "FALSE": 63 refclk period "TRUE": 31 refclk period
PMA_REG_I_CTRL_MAX	<integer>	[63, 0]	45	Variable resistance control code maximum value (feedback to variable resistance)
PMA_REG_I_CTRL_MIN_1_0	<integer>	[3, 0]	3	Together form the minimum value of the variable resistor control code (feedback to the variable resistor)
PMA_REG_I_CTRL_MIN_5_2	<integer>	[15, 0]	4	
PMA_REG_RESCAL_I_CODE_3_0	<integer>	[15, 0]	0	Together they form the initial value of the variable resistor control code and the unsuccessful default value
PMA_REG_RESCAL_I_CODE_5_4	<integer>	[3, 0]	2	
PMA_REG_RESCAL_INT_R_SMALL_VAL	<string>	"FALSE"; "TRUE"	"FALSE"	On-chip resistance is small, reg signal
PMA_REG_RESCAL_INT_R_SMALL_OW	<string>	"FALSE"; "TRUE"	"FALSE"	Small on-chip resistor selection control signal Register overwrite enable for rescal_int_r_small_i "FALSE": no overwrite "TRUE": port is overwriten by reg_rescal_int_r_small

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_RESCAL_ I_CODE_PMA	<string>	"FALSE"; "TRUE"	"FALSE"	The two registers reg_rescal_i_code_ow_i and reg_rescal_i_code_pma_i are 1x: pma_rescal_i_code_o is reg_rescal_i_code_val_i; 01: pma_rescal_i_code_o is pma_rescal_i_code_i; 00: pma_rescal_i_code_o is rescal_i_code. rescal_i_code is an internal signal.
PMA_REG_REFCLK0_ JTAG_OE	<string>	"FALSE"; "TRUE"	"FALSE"	refclk0_jtag_oe
PMA_REG_REFCLK1_ JTAG_OE	<string>	"FALSE"; "TRUE"	"FALSE"	refclk1_jtag_oe
PMA_REG_RES_CAL_EN	<string>	"FALSE"; "TRUE"	"FALSE"	Resistance Calibration Enable
PMA_REG_HPLL_RSTN	<string>	"FALSE"; "TRUE"	"FALSE"	hpll_rstn
PMA_REG_HPLL_RSTN_OW	<string>	"FALSE"; "TRUE"	"FALSE"	hpll_rstn_ow
PMA_REG_HPLL_PD	<string>	"FALSE"; "TRUE"	"FALSE"	hpll power down
PMA_REG_HPLL_PD_OW	<string>	"FALSE"; "TRUE"	"FALSE"	hpll power down overwrite
PMA_REG_LC_VCO_ CAL_EN	<string>	"FALSE"; "TRUE"	"FALSE"	Enabling of lc_vco as lockdetect reset source
PMA_REG_DIV_CALI_ BYPASS	<string>	"TRUE"; "FALSE"	"TRUE"	div_cali_bypass Fixed to "TRUE"
PMA_REG_CALIB_WAIT	<integer>	[3, 0]	3	Feedback clock wait time configuration from reg_reg_calib_wait: 0: wait period=13'd1023; 1: waiting period = 13'd2047; 2: Waiting period = 13'd4095; 3: Waiting period = 13'd511; Waiting time in CALIB_WAIT state
PMA_REG_CALIB_TIMER	<integer>	[3, 0]	0	from reg_reg_calib_timer Feedback clock statistics time selection configuration: 0: statistical period=13'd511; 1: statistical period=13'd1023; 2: statistical period=13'd2047; 3: Statistics period=13'd4095.
PMA_REG_BAND_LB	<integer>	[31, 0]	0	The window start value configuration point of the frequency adjustment band
PMA_REG_BAND_HB_0	<string>	"TRUE"; "FALSE"	"TRUE"	Frequency adjustment band window end value configuration point

Parameter Name	Parameter Type	Valid value	Defaults	Functional description
PMA_REG_BAND_HB_4_1	<integer>	[15, 0]	15	Frequency adjustment band window end value configuration point
PMA_CFG_HSST_RSTN	<string>	"TRUE"; "FALSE"	"TRUE"	Register configuration hsst_rstn
PMA_CFG_PLLPOWERUP	<string>	"OFF"; "ON"	"ON"	Register configuration pllpowerup
PMA_PLL_RSTN	<string>	"TRUE"; "FALSE"	"TRUE"	Register configuration pll_rstn
PMA_REG_IBIAS_STATIC_SEL_7_0	<integer>	[255, 0]	255	reg_ibup_dyn_a<8:1> ibias static current selection: 0: 1X/3X 1: 2X/4X (1X current=25uA)
PMA_REG_IBIAS_STATIC_SEL_15_8	<integer>	[255, 0]	247	reg_ibup_dyn_a<16:9> ibias static current selection: 0: 1X/3X 1: 2X/4X (1X current=25uA)
PMA_REG_IBIAS_STATIC_SEL_18_16	<integer>	[7, 0]	7	reg_ibup_dyn_a<19:17> ibias static current selection: 0: 1X/3X 1: 2X/4X (1X current=25uA)
PMA_REG_IBIAS_STATIC_SEL_21_19	<integer>	[7, 0]	0	reg_ibup_dyn_b<3:1> ibias static current selection: 0: 1X/2X 1: 3X/4X (1X current=25uA)
PMA_REG_IBIAS_STATIC_SEL_29_22	<integer>	[255, 0]	0	reg_ibup_dyn_b<11:4> ibias static current selection: 0: 1X/2X 1: 3X/4X (1X current=25uA)
PMA_REG_IBIAS_STATIC_SEL_37_30	<integer>	[255, 0]	1	reg_ibup_dyn_b<19:12> ibias static current selection: 0: 1X/2X 1: 3X/4X (1X current=25uA)
PMA_REG_HPLL_LDO_VOL_RANGE_O	<string>	"943MV"; "960MV"; "978MV"; "1000MV"; "1023MV"; "1050MV"; "1080MV"; "DISABLE"	"1000MV"	Set the ldo output voltage amplitude
PMA_REG_HPLL_VCTRL_SEL_L	<string>	"0.1V"; "0.2V"; "0.3V"; "0.4V"	"0.4V"	Resistor selection to adjust output vctrl voltage

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
PMA_REG_HPLL_ VCTRL_SEL_H	<string>	"0.5V"; "0.6V"; "0.7V"; "0.8V"	"0.5V"	Resistor selection to adjust output vctrl voltage
PMA_REG_HPLL_ PCURRENT_SEL	<string>	"16*IO"; "8*IO"; "12*IO_0"; "12*IO_1"	"16*IO"	P tube bias current adjustment
PMA_REG_HPLL_ PCURRENT_SEL0	<string>	"8*IO"; "4*IO"; "6*IO"; "10*IO"	"10*IO"	N tube bias current adjustment
PMA_REG_IBIAS_ STA_CUR_SEL_4_0	<integer>	[31, 0]	31	reg_ibup_sta_a<5:1> ibias static current selection: 0: 1X/3X 1: 2X/4X (1X current=12.5uA)
PMA_REG_IBIAS_ STA_CUR_SEL_8_5	<integer>	[15, 0]	15	reg_ibup_sta_a<9:6> ibias static current selection: 0: 1X/3X 1: 2X/4X (1X current=12.5uA)
PMA_REG_IBIAS_ STA_CUR_SEL_12_9	<integer>	[15, 0]	0	reg_ibup_sta_b<4:1> ibias static current selection: 0: 1X/2X 1: 3X/4X (1X current=12.5uA)
PMA_REG_IBIAS_ STA_CUR_SEL_17_13	<integer>	[31, 0]	0	reg_ibup_sta_b<9:5> ibias static current selection: 0: 1X/2X 1: 3X/4X (1X current=12.5uA)
PMA_REG_HPLL_LDO_ VOL_RANGE_I	<string>	"640MV"; "534MV"	"640MV"	ldo input voltage amplitude adjustment

5. Instantiation template

```
GTP_HSSTHP_HPLL #(
.PMA_REG_REFCLK0_IMPEDANCE_SEL("100_OHM"),
.PMA_REG_REFCLK1_IMPEDANCE_SEL("100_OHM"),
.PMA_REG_PLL_JTAG0_VTH_SEL("60MV"),
.PMA_REG_PLL_JTAG0_LPF_RSEL("20K"),
.PMA_REG_PLL_JTAG1_VTH_SEL("60MV"),
.PMA_REG_PLL_JTAG1_LPF_RSEL("20K"),
.PMA_REG_IBIAS_STATIC_SEL_7_0(255),
```

```
.PMA_REG_IBIAS_STATIC_SEL_15_8(247),
.PMA_REG_IBIAS_STATIC_SEL_18_16(7),
.PMA_REG_IBIAS_STATIC_SEL_21_19(0),
.PMA_REG_IBIAS_STATIC_SEL_29_22(0),
.PMA_REG_IBIAS_STATIC_SEL_37_30(1),
.PMA_REG_IBIAS_DYNAMIC_PD_7_0(0),
.PMA_REG_IBIAS_DYNAMIC_PD_15_8(0),
.PMA_REG_IBIAS_DYNAMIC_PD_18_16(0),
.PMA_REG_IBIAS_STA_CUR_SEL_4_0(31),
.PMA_REG_IBIAS_STA_CUR_SEL_8_5(15),
.PMA_REG_IBIAS_STA_CUR_SEL_12_9(0),
.PMA_REG_IBIAS_STA_CUR_SEL_17_13(0),
.PMA_REG_IBIAS_STA_CUR_PD_2_0(0),
.PMA_REG_IBIAS_STA_CUR_PD_8_3(0),
.PMA_REG_BANDGAP_VOL_SEL("BANDGAP"),
.PMA_REG_BANDGAP_TEST(0),
.PMA_REG_CALIB_CLKDIV_RATION("DIV1"),
.PMA_REG_TX_RATE_CHANGE_SEL0("CLK_FROM_HPLL"),
.PMA_REG_TX_RATE_CHANGE_SEL1("SEL_SYNC_RATE_CHANGE"),
.PMA_REG_TLING_IMPEDANCE_CTRL("125OHM"),
.PMA_ANA_COM_REG_143(0),
.PMA_ANA_COM_REG_149(0),
.PMA_ANA_COM_REG_155(0),
.PMA_REG_HPLL_DIV_CHANGE_3_0(0),
.PMA_REG_HPLL_DIV_CHANGE_11_4(0),
.PMA_REG_HPLL_DIV_CHANGE_15_12(0),
.PMA_REG_HPLL_VCOLDO_EN("TRUE"),
.PMA_REG_HPLL_LDO_VOL_RANGE_O("1000MV"),
.PMA_REG_HPLL_LDO_VOL_RANGE_I("640MV"),
.PMA_REG_HPLL_REFCLK_DIV(1),
.PMA_REG_HPLL_CHARGE_PUMP_PD("WORK"),
.PMA_REG_HPLL_CHARGE_PUMP_CTRL(10),
.PMA_REG_HPLL_VCTRL_SEL_L("0.4V"),
.PMA_REG_HPLL_VCTRL_SEL_H("0.5V"),
.PMA_REG_HPLL_LPF_RES_SEL("5K"),
.PMA_REG_HPLL_PCURRENT_SEL("16*IO"),
.PMA_REG_HPLL_VCO_EN(1),
.PMA_REG_HPLL_PCURRENT_SEL0("10*IO"),
.PMA_REG_HPLL_PCURRENT_SEL1(0),
.PMA_REG_REFCLK_RST_SEL(0),
.PMA_REG_NFC_STIC_DIS_N(0),
.PMA_REG_HPLL_FBDIV0(20),
.PMA_REG_HPLL_FBDIV1(20),
.PMA_REG_HPLL_PHASE_SEL("DIV2"),
.PMA_REG_HPLL_CFG_7_0(7),
.PMA_REG_HPLL_CFG_15_8(132),
.PMA_REG_PLL_LOCKDET_RESET_N("FALSE"),
.PMA_REG_PLL_LOCKDET_RESET_N_OW("FALSE"),
```

```
.PMA_REG_READY_OR_LOCK("FALSE"),
.PMA_REG_HPLL_READY("FALSE"),
.PMA_REG_HPLL_READY_OW("FALSE"),
.PMA_REG_PLL_LOCKDET_REFCT_0("TRUE"),
.PMA_REG_PLL_LOCKDET_REFCT_2_1(1),
.PMA_REG_PLL_LOCKDET_FBCT(3),
.PMA_REG_PLL_LOCKDET_LOCKCT(4),
.PMA_REG_PLL_LOCKDET_ITER(0),
.PMA_REG_PLL_UNLOCKDET_ITER(2),
.PMA_REG_PLL_LOCKDET_EN_OW("FALSE"),
.PMA_REG_PLL_LOCKDET_EN("FALSE"),
.PMA_REG_PLL_LOCKDET_MODE("FALSE"),
.PMA_REG_PLL_LOCKED_OW("FALSE"),
.PMA_REG_PLL_LOCKED("FALSE"),
.PMA_REG_PLL_UNLOCKED_OW("FALSE"),
.PMA_REG_PLL_UNLOCKED("FALSE"),
.PMA_REG_PLL_LOCKED_STICKY_CLEAR("FALSE"),
.PMA_REG_PLL_UNLOCKED_STICKY_CLEAR("FALSE"),
.PMA_REG_LOCKDET_REPEAT("FALSE"),
.PMA_REG_NOFBCLK_STICKY_CLEAR("FALSE"),
.PMA_REG_NOREFCLK_STICKY_CLEAR("FALSE"),
.PMA_REG_RESCAL_EN("FALSE"),
.PMA_REG_RESCAL_RST_N_OW("FALSE"),
.PMA_REG_RESCAL_RST_N_VAL("FALSE"),
.PMA_REG_RESCAL_DONE_VAL("FALSE"),
.PMA_REG_RESCAL_DONE_OW("FALSE"),
.PMA_REG_RESCAL_I_CODE_VAL_1_0(0),
.PMA_REG_RESCAL_I_CODE_VAL_5_2(0),
.PMA_REG_RESCAL_I_CODE_OW("FALSE"),
.PMA_REG_RESCAL_ITER_VALID_SEL(0),
.PMA_REG_RESCAL_WAIT_SEL("FALSE"),
.PMA_REG_I_CTRL_MAX(45),
.PMA_REG_I_CTRL_MIN_1_0(3),
.PMA_REG_I_CTRL_MIN_5_2(4),
.PMA_REG_RESCAL_I_CODE_3_0(0),
.PMA_REG_RESCAL_I_CODE_5_4(2),
.PMA_REG_RESCAL_INT_R_SMALL_VAL("FALSE"),
.PMA_REG_RESCAL_INT_R_SMALL_OW("FALSE"),
.PMA_REG_RESCAL_I_CODE_PMA("FALSE"),
.PMA_REG_REFCLK0_JTAG_OE("FALSE"),
.PMA_REG_REFCLK1_JTAG_OE("FALSE"),
.PMA_REG_RES_CAL_EN("FALSE"),
.PMA_REG_HPLL_RSTN("FALSE"),
.PMA_REG_HPLL_RSTN_OW("FALSE"),
.PMA_REG_HPLL_PD("FALSE"),
.PMA_REG_HPLL_PD_OW("FALSE"),
.PMA_REG_LC_VCO_CAL_EN("FALSE"),
.PMA_REG_DIV_CALI_BYPASS("TRUE"),
```

```
.PMA_REG_CALIB_WAIT(3),
.PMA_REG_CALIB_TIMER(0),
.PMA_REG_BAND_LB(0),
.PMA_REG_BAND_HB_0("TRUE"),
.PMA_REG_BAND_HB_4_1(15),
.PMA_CFG_HSST_RSTN("TRUE"),
.PMA_CFG_PLLPOWERUP("ON"),
.PMA_PLL_RSTN("TRUE")
) <InstanceName> (
.P_CFG_RDATA_HPLL(),            // OUTPUT[7:0]
.P_RES_CAL_CODE_FABRIC(),       // OUTPUT[5:0]
.P_CFG_ADDR_HPLL(),             // INPUT[11:0]
.P_CFG_WDATA_HPLL(),            // INPUT[7:0]
.P_RESCAL_I_CODE_I(),           // INPUT[5:0]
.DIV_SYNC_REFSYNC_O(),          // OUTPUT
.PMA_HPLL_CK0(),                // OUTPUT
.PMA_HPLL_CK90(),               // OUTPUT
.PMA_HPLL_CK180(),              // OUTPUT
.PMA_HPLL_CK270(),              // OUTPUT
.PMA_HPLL_READY_O(),            // OUTPUT
.PMA_HPLL_REFCLK_O(),           // OUTPUT
.PMA_TX_SYNC_HPLL_O(),          // OUTPUT
.P_CFG_INT_HPLL(),              // OUTPUT
.P_CFG_READY_HPLL(),            // OUTPUT
.P_HPLL_READY(),                // OUTPUT
.REFCLK_SYNC_REFSYNC_O(),       // OUTPUT
.TX_SYNC_REFSYNC_O(),           // OUTPUT
.ANA_HPLL_DIV_SYNC_I(),         // INPUT
.ANA_HPLL_REFCLK_SYNC_I(),      // INPUT
.ANA_TX_SYNC_I(),               // INPUT
.P_CFG_CLK_HPLL(),              // INPUT
.P_CFG_ENABLE_HPLL(),           // INPUT
.P_CFG_PSEL_HPLL(),             // INPUT
.P_CFG_RST_HPLL(),              // INPUT
.P_CFG_WRITE_HPLL(),            // INPUT
.P_HPLL_DIV_CHANGE(),           // INPUT
.P_HPLL_DIV_SYNC(),             // INPUT
.P_HPLL_LOCKDET_RST(),          // INPUT
.P_HPLL_POWERDOWN(),            // INPUT
.P_HPLL_REFCLK_I(),             // INPUT
.P_HPLL_RST(),                  // INPUT
.P_HPLL_VCO_CALIB_EN(),         // INPUT
.P_PLL_REFCLK6_I(),             // INPUT
.P_REFCLK_DIV_SYNC(),           // INPUT
.P_RES_CAL_RST(),               // INPUT
.P_TX_SYNC(),                   // INPUT
.REFCLK_TO_DIV_SYNC_I(),        // INPUT
.REFCLK_TO_REFCLK_SYNC_I(),     //  INPUT
```

```
.REFCLK_TO_TX_SYNC_I()          // INPUT
);
```

(9) GTP_HSSTHP_BUFDS instruction

1. Supported devices

Table 355 GTP_HSSTHP_BUFDS device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

Inside HSSTHP, each Quad has 2 HSSTHP_BUFDS, which can be used at the same time. The PDS software will automatically map according to the physical position of the reference clock PAD connected to the port PAD_REFCLKP and PAD_REFCLKN. When the user restricts to use REFCLK0, the software will map the corresponding BUFDS to HSSTHP_BUFDS_0; when the user restricts to use REFCLK1, the software will map the corresponding BUFDS to HSSTHP_BUFDS_1.

3. Port description

Table 356 GTP_HPIO_VREF port list description

Port Name	Direction	Width	Functional description
COM_POWERDOWN	I	1	The powerdown control port of pma_common, 1: power down; 0: power on
PAD_REFCLKP	I	1	Differential Reference Clock Input Port (Positive)
PAD_REFCLKN	I	1	Differential Reference Clock Input Port (Negative)
REFCLK_OUTP	O	1	Single-ended reference clock output port, the output is sent to HPLL or LANE
PMA_REFCLK_TO_FABRIC	O	1	Output the reference clock sent to the fabric, you can pass the parameter PMA_REG_REFCLK_FAB configures this output signal

4. Parameter description

Table 357 GTP_HPIO_VREF parameter list

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
PMA_CFG_COMMPOWERUP	<string>	"OFF"; "ON"	"ON"	Register configuration commpowerup
PMA_REG_COM_PD	<string>	"FALSE"; "TRUE"	"FALSE"	configure com powerdown
PMA_REG_COM_PD_OW	<string>	"FALSE"; "TRUE"	"FALSE"	configure com powerdown overwrite
PMA_REG_HPLL_REFCLK_PD	<string>	"ON"; "OFF"	"ON"	HPLL reference clock powerdown control
PMA_REG_REFCLK_FAB	<string>	"REFCLK"; "REFCLK/2";	"REFCLK"	Configure the reference clock output to the fabric; "REFCLK": refclk;

		"DISABLE"		"REFCLK/2": refclk/2; "DISABLE": disable.

5. Instantiation template

```
GTP_HSSTHP_BUFDS #(
.PMA_CFG_COMMPOWERUP("ON"),
.PMA_REG_COM_PD("FALSE"),
.PMA_REG_COM_PD_OW("FALSE"),
.PMA_REG_HPLL_REFCLK_PD("ON"),
.PMA_REG_REFCLK_FAB("REFCLK")
) GTP_HSSTHP_BUFDS_inst (
.PMA_REFCLK_TO_FABRIC(),// OUTPUT
.REFCLK_OUTP(), // OUTPUT
.COM_POWERDOWN(), // INPUT
.PAD_REFCLKN(), // INPUT
.PAD_REFCLKP() // INPUT
);
```

(10) GTP_HPIO_VREF instruction

1. Supported devices

Table 358 GTP_HPIO_VREF device support

Device family	Titan2
Whether to support the GTP	Support

2. Function introduction

GTP_HPIO_VREF is responsible for providing reference voltage to GPIO, and its structural block diagram is shown below.

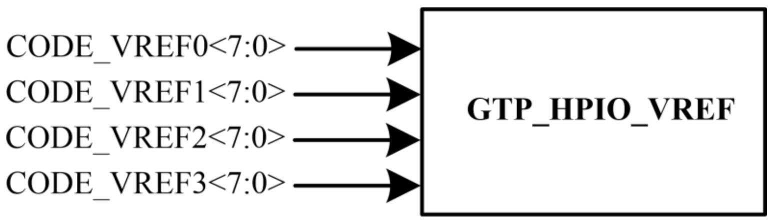


Figure 123 GTP_HPIO_VREF block diagram

3. Port description

Table 359 GTP_HPIO_VREF port list description

Port Name	Direction	Functional description
CODE_VREF0<7:0>	Input	Group0 internal reference voltage calibration code value
CODE_VREF1<7:0>	Input	Group1 internal reference voltage calibration code value
CODE_VREF2<7:0>	Input	Group2 internal reference voltage calibration code value

CODE_VREF3<7:0>	Input	Group3 internal reference voltage calibration code value

4. Parameter description

Table 360 GTP_HPIO_VREF parameter list

Parameter Name	Parameter Type	Effective value	Defaults	Functional description
BANK_LOC	String	BANKR5/ BANKR6/ BANKR7	BANKR5	Specify BANK location

5. Instantiation template

```
GTP_HPIO_VREF #(
.BANK_LOC("BANKR5")
) <InstanceName> (
.CODE_VREF0(),// INPUT[7:0]
.CODE_VREF1(),// INPUT[7:0]
.CODE_VREF2(),// INPUT[7:0]
.CODE_VREF3() // INPUT[7:0]
);
```

Disclaimer

Copyright Notice

Disclaimer

1. This document only provides phased information, and the content contained in it can be updated at any time according to the actual situation of the product without prior notice. The company does not assume any legal responsibility for any direct or indirect losses caused by improper use of this document.

2. This document is provided as is, without any guarantee, including any guarantee of merchantability, suitability for a particular purpose, or non-infringement, and any guarantee mentioned elsewhere in any proposal, specification or sample. No license, express or implied, to any intellectual property right is hereby granted by this document, whether by estoppel or otherwise.

3. The company reserves the right to modify the relevant documents of the company's series of products at any time without prior notice.

Замечания по переводу отправляйте в ЛС пользователю Zig сайта electronix.ru